The Dissertation Committee for Sara Kathryn Scarritt
certifies that this is the approved version of the following dissertation:

A Self-Contained Guidance and Targeting Algorithm for Spacecraft Applications

Committee:

Belinda G. Marchand, Supervisor

David G. Hull

Cesar A. Ocampo

Christopher N. D'Souza

Michael W. Weeks

A Self-Contained Guidance and Targeting Algorithm for Spacecraft Applications

by

Sara Kathryn Scarritt, B.S.E., M.S.AS.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2012

For my parents, who have always believed in me.

Acknowledgments

In my academic career, I have been incredibly fortunate to be surrounded by truly wonderful mentors, colleagues, and friends. I am so thankful to God for introducing me to these amazing people, and for all the opportunities He has given me - even (and perhaps especially) the ones I didn't recognize as opportunities at first.

First and foremost, I want to thank my advisor, Dr. Belinda Marchand. Her guidance, expertise, and motivation have been invaluable throughout this process, and I am a better engineer and a better person for having studied under her instruction. I am also grateful to my committee members, Dr. David Hull, Dr. Cesar Ocampo, Dr. Chris D'Souza, and Dr. Mike Weeks, for all that they have taught me inside the classroom/workplace, and for all of their support and encouragement outside of it.

I owe a huge debt of gratitude to my friends and colleagues at the Air Force Research Lab, particularly Adam Fosbury, Deb Fogle, and Robbie Robertson. Their support helped me continue my graduate education when I didn't think I could, and their advice led me here to UT, and I cannot thank them enough for that. A big thank you also goes to the folks at Wash U: Dr. Mike Swartwout, Stephen Forbes, and the entire Bandit/Akoya team. It was a joy to work with you all, and I can't wait to see Bandit fly!

To my UT family - my lab mates; the incomparable Aerospace Girls Squad; and especially my past and present office mates, Chad Smith, Stuart Stanton, Andrew Takano, and Divya Thakur - thank you so much for all your help, for being sounding boards whenever I got stuck, and above all for your friendship. You've made Austin a home rather than just a place where I live. Special thanks to Divya, Andrew, and Sonia Hernandez for taking the time to read over this dissertation and provide feedback.

Finally, and most importantly, I want to thank my family. Their unconditional love, support, and inspiration has carried me always, and I could not have done this without them.

A Self-Contained Guidance and Targeting Algorithm for Spacecraft Applications

Sara Kathryn Scarritt, Ph.D.
The University of Texas at Austin, 2012

Supervisor: Belinda G. Marchand

The development of a self-contained, onboard, fully autonomous trajectory guidance tool for spacecraft is presented. To be considered completely autonomous requires the capability to both identify an appropriate startup solution, and then use that solution to target a set of user-defined path and endpoint constraints. To minimize the cost of flight software development and validation, both the generation of the startup solution and the targeting algorithm are designed to be as computationally efficient as possible. This study addresses both the determination of a startup arc and the subsequent targeting process.

The first part of the investigation considers the targeting algorithm. Linear targeting through differential corrections is a well-known approach for identifying feasible solutions that meet specified mission and trajectory constraints. However, to date, these methods relied on the assumption that the associated control inputs were impulsive in nature. This research focuses on

the theoretical development and numerical validation of a generalized linear targeting algorithm capable of accommodating finite periods of continuous control action for a wide range of applications. Examples are presented to illustrate the general concept and to contrast the performance of this new targeting process against more classical impulsive targeting methods.

The second section of the study introduces a novel approach utilizing artificial potential function methods to identify suitable startup solutions. Although common in other types of path planning, these methods have not yet been used for orbital or interplanetary trajectory design, primarily due to their inherent suboptimality. However, results show that this issue can be addressed with relative ease by the targeting algorithm.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Spacecraft autonomy is a subject of continuing interest across many different research disciplines. In general, autonomy refers to the ability of a vehicle or machine to achieve a desired series of objectives without requiring human interaction. More specifically, within the context of this investigation, autonomy is defined as the ability to (a) automatically identify a suitable startup trajectory[15, 34, 62] and (b) use that solution to successfully target some specified set of terminal and path constraints within the actuation budget available at the time[25]. Targeting, in this study, is the process of computing the initial control parameters, including start time and duration, if applicable, required to reach a desired end condition.

The first part of this study addresses the second stage of the autonomous targeting process, re-targeting all of the constraints using only the resources available onboard at the time (e.g. fuel left and operational engine, for a spacecraft performing orbital maneuvers) based on the initial guess supplied. Specifically, Chapter 3 presents a robust and computationally efficient algorithm for targeting a feasible trajectory using continuous actuation applied over some finite time period. While the problem of impulsive targeting has

been studied at length[25, 58, 61], there are certain scenarios, depending on the actuation method, in which it is no longer accurate to approximate each maneuver as impulsive. Thus, the quality of an initial guess that assumes impulsive maneuvers degrades significantly. Furthermore, many dynamical systems assume a continuous control input over specified time intervals, and so an algorithm that can accommodate this type of actuation is required in order to target trajectories for these systems.

The second part of the investigation focuses on the identification of a valid startup arc. The success of iterative gradient based targeting algorithms, whether optimal or suboptimal, depends on the quality of the startup solution available. Thus, autonomous targeting[58] suggests autonomy in both the computational process used to identify a feasible or optimal solution, given some initial guess, and in the process of identifying the startup arc itself. Regardless of the targeting algorithm selected, the startup arc need not be completely feasible. However, the quality of the startup solution does influence the performance of gradient based targeting algorithms. Furthermore, the solution space explored by gradient based targeting methods is restricted to the immediate vicinity of the startup arc. This may ultimately limit the types of arcs identified by the targeting process.

The final phase of this study utilizes artificial potential function (APF) methods as a tool for generating suitable startup solutions. They are computationally efficient and simple to implement, making them ideal for onboard use, but they also have inherent limitations in that (1) a continuous control

capability is typically assumed and (2) they are also often highly suboptimal. These drawbacks hinder the effectiveness of potential function guidance as a stand alone trajectory design tool, since actuation constraints are not uncommon in modern spacecraft missions. However, both of these issues can, in theory, be addressed by autonomous targeters and optimizers, such as the algorithm described in Chapter 3. Thus, potential function methods are explored as a means of quickly and efficiently identifying the startup arc, while autonomous targeting algorithms are employed to impose any actuator or mission constraints in the final trajectory.

1.1 Previous Work

1.1.1 Continuous Actuation Targeting

For general systems with continuous actuation, the available literature focuses almost exclusively on control algorithms, rather than targeting. The planning and execution of attitude maneuvers, for example, is often approached as either an adaptive control problem[1, 20, 64] or an optimization problem[6, 23]. In the field of spacecraft trajectory design, where impulsive inputs are typically assumed, the majority of the research on finite thrust guidance has again been concentrated on finding optimal trajectories[9, 18, 32, 41], rather the feasible ones. Among these methods, nonlinear programming is commonly employed in solving optimal and nonlinear targeting problems[32, 41, 65]. The process of numerically identifying optimal or feasible solutions via nonlinear programming is basically the same. The main

difference between targeting and optimization is that optimization problems require that a cost index be specified and feasibility problems, such as constrained nonlinear targeting, do not. Of course, the identification of feasible solutions that meet all the specified constraints is also accomplished through linear targeting methods[25, 61]. These classical methods also use the state transition matrix to compute the necessary constraint gradients, but these are then implemented in a linear corrections process. Naturally, a nonlinear process is preferred when the computational resources are available. However, for onboard determination, the optimality of a solution is not as critical as the availability of a feasible solution. In this case, the inherent simplicity of linear targeting algorithms leads to a reduced cost in flight software development and validation.

Some studies consider optimization methods for use during onboard targeting processes. These include the use of a simplified adaptive guidance law for targeting relative to a predetermined nominal trajectory[52] or implementation of an efficient sequential gradient-restoration algorithm employing multiple subarcs[30]. For attitude maneuver planning, randomized search algorithms have been utilized to find feasible trajectories[8, 19]. These methods, however, still employ some measure of optimization to find a solution. A more computationally efficient method, commonly used for missile guidance and spacecraft rendezvous, is to solve Lambert's problem to recover the initial velocity required to reach a specified final position in a given time-of-flight. Because the Lambert solution does not take gravitational perturbations

into account, this approach can result in final position errors, particularly in spacecraft applications. To counteract this issue, the Space Shuttle's Lambert targeting algorithm uses an iterative loop with error offsetting to converge on the correct initial velocity[29]. The Lambert solution can also be used for non-impulsive missiles or vehicles; Lambert's problem is solved at each time instant during powered flight, and the vector error between the current velocity and the computed desired velocity is used to calculate a desired thrust direction. This process is known as Lambert guidance[63]. The targeting algorithm developed in this study incorporates ideas from both Lambert guidance and linear targeting methods.

1.1.2 Initial Guess Determination

Preliminary trajectory design is often accomplished through patched-conic approximations[39]. Of course, two-body patched-conic solutions propagated in a more complete n-body model, for $n \geq 3$, do not lead to feasible trajectory arcs since a discontinuity at the patch-point is generally expected. Iterative targeting algorithms, optimal or suboptimal, are necessary to reacquire a feasible continuous solution in the full dynamical model. Ultimately, the accuracy of the patched-conic approximation varies according to the regime and any related mission requirements. Low-energy spacecraft trajectories that dwell near the gravitational boundary of two bodies, for instance, are more susceptible to third-body effects. Recent studies also investigate the use of graphical methods[36, 48] and extensions of the patched-conic approach to in-

clude approximations of three-body motion.[61] However, these approaches are typically intended for specific types of transfers, such as low-thrust and/or gravity assist trajectories. The main drawback of any methodology based on patched-conic approximations is that one cannot guarantee the resulting arcs will lead to reasonable approximations in multi-body regimes during an onboard determination process.

For more general path planning, artificial potential functions have been used extensively for a variety of applications. Early use of this method is predominantly in the field of robot motion planning[43], but in recent years the approach has been applied to on-orbit applications such as attitude control[26, 28] and formation flight[27, 31, 50, 51]. Methodologies developed for potential function control of formation flight is of particular interest in this study, as these previous investigations consider the issue of impulsive and finite burn control inputs[27]. References [31] and [50] describe the construction of desired velocity fields and a velocity error-based potential function, both of which can be modified for use in general trajectory design.

1.2 Method Overview

1.2.1 Continuous Actuation Targeting

The solution process behind the algorithm presented is partly modeled after the two-level targeter[12, 13, 25, 59–61], employed during the design of the Genesis trajectory[12, 60, 61]. However, unlike this earlier development, the present algorithm allows for the incorporation of finite periods of continu-

ous actuation. A two-level targeter (or corrector) is primarily based on linear system theory; it uses a time-varying linearized dynamical model and a minimum norm solution to compute solution updates. These linear updates are implemented in the nonlinear system in an iterative corrections process that repeats until a feasible solution is identified in the vicinity of the startup arc. The two-level process offers several advantages: because the updates are based on the linearized model, it is numerically simple and computationally efficient. It does not require knowledge of a nominal solution, relying instead solely on the current path of the vehicle. The two-level corrections process also allows for straightforward addition of path constraints, both those at specific points (e.g. entry interface) [25, 59–61] and those applied over the trajectory as a whole [59]. However, it was originally designed to use impulsive maneuvers as control variables. In this investigation, the classical impulsive two-level corrections process[25] is modified to incorporate actuator dynamics while still retaining the structure and simplicity of the original algorithm so that it is suitable for onboard calculations. This is achieved by first augmenting the vehicle state vector to include actuator parameters, and then dividing each trajectory arc into "on/off" subarcs, depending on whether the control input is active. These modifications allow the both the nonlinear and linearized models to retain the same general form used in the derivation of the classical two-level corrector.

1.2.2 Initial Guess Determination

Artificial potential function guidance is based on the idea that the vehicle environment can be represented mathematically through the definition of potential fields designed to produce some desired vehicle behavior. The general approach is to construct the potential field such that a global minimum exists at the target state while any path constraints, such as obstacles, are assigned higher potentials to discourage the vehicle from traversing those paths. The potential, P, is written as a nonlinear function of the current position and velocity, $\mathbf{r}(t)$ and $\mathbf{v}(t)$, the desired position and velocity, $\mathbf{r}_{des}$ and $\mathbf{v}_{des}$, and any existing constraints (γ):

$$P = f\left(\mathbf{r}(t), \mathbf{v}(t), \mathbf{r}_{des}, \mathbf{v}_{des}, \gamma\right) \tag{1.1}$$

Control variables and parameters are subsequently selected such that the vehicle follows the path of steepest descent of the potential. If no local minima exist, convergence to some desired, or "goal", state is ensured. In spacecraft applications, potential function methods are used primarily for planning relative motion within a formation or attitude trajectories. However, the theoretical basis of this method does not preclude the use of potential functions for more general trajectory design. The key is the construction of the potential. In this study, two-body approximations are utilized as a first step towards constructing a set of artificial potentials that facilitate the identification of trajectories with the desired characteristics. The insight acquired through this analysis is subsequently useful in maneuver planning for the associated startup arc.

1.3 Organization

This work is organized as follows:

- **Chapter 2: Background** The relevant theoretical background is established. First, linear targeting methods are reviewed, with particular emphasis on the development of the two-level correction process. Next, an overview of artificial potential function methods is given. The advantages and disadvantages of these methods are discussed in order to justify the chosen approach.

- **Chapter 3: Continuous Actuation Targeting Algorithm** The development of a continuous actuation targeting algorithm is described in detail. First, the general dynamics for a continuous actuation problem are discussed in relation to the two-level corrector framework in order to highlight the issues that arise from the introduction of continuous actuation. Modifications are then applied to both the classical two-level correction process and the dynamics representation to address these issues. Derivation and application examples are provided for selected actuation methods.

- **Chapter 4: Initial Guess Determination** An initial guess algorithm based on Artificial Potential Function (APF) methods is presented. Procedures for constructing different artificial potential fields are discussed. These fields are used independently or in combinations to produce some

desired behavior by the vehicle. Intermediate results are presented for orbital and interplanetary transfers.

- **Chapter 5: End-to-End Examples** Several examples are presented to demonstrate the targeting and initial guess algorithms working together in tandem. Emphasis is placed on the complementary nature of the two algorithms, particularly the ability of each one to satisfactorily address any weaknesses in the other. Results are also examined to determine areas for possible future research efforts.

- **Chapter 6: Conclusions** A summary of this effort is provided. Recommendations are discussed for future work that may enhance the current method.

Chapter 2

Background

2.1 Linear Targeting Methods

2.1.1 Linearized Dynamic Model

The development and implementation of a linear targeting algorithm depends on the availability of a linearized model to represent the nonlinear system dynamics. Consider a general zero-input nonlinear system whose motion is described by the differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t). \tag{2.1}$$

Let $\tilde{\mathbf{x}}(t)$ be a neighboring trajectory solution such that $\dot{\tilde{\mathbf{x}}}(t) = \mathbf{f}(\tilde{\mathbf{x}}(t), t)$ and

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t) + \delta\mathbf{x}(t), \tag{2.2}$$

where $\delta\mathbf{x}(t)$ is a contemporaneous variation - that is, it is the difference between the new path $\tilde{\mathbf{x}}(t)$ and the current path $\mathbf{x}(t)$ evaluated exactly at time t. If the time is varied as well, such that $t = t' + dt$, then $\tilde{\mathbf{x}}(t)$ can be written as

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t') + d\mathbf{x}(t). \tag{2.3}$$

The vector $d\mathbf{x}(t)$ is the non-contemporaneous variation, or differential, between $\tilde{\mathbf{x}}(t)$ and $\mathbf{x}(t')$. A graphical representation of the relationship between $\delta\mathbf{x}$ and $d\mathbf{x}$ is given in Figure 2.1.

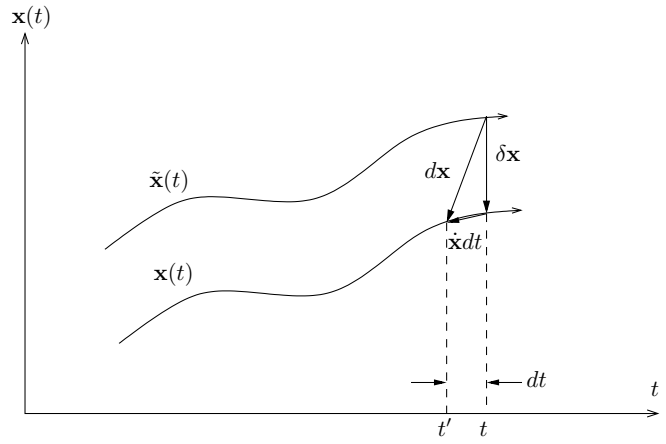


Figure 2.1: Variation vs. Differential

Replacing $\mathbf{x}(t') = \mathbf{x}(t - dt)$ with a Taylor Series approximation about $\mathbf{x}(t)$ gives

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \left[\frac{d\mathbf{x}}{dt}\bigg|_t dt + \frac{1}{2!}\frac{d^2\mathbf{x}}{dt^2}\bigg|_t d^2t + ...\right] + d\mathbf{x}(t). \tag{2.4}$$

A mathematical expression for the relationship between the variation $\delta\mathbf{x}$ and the differential $d\mathbf{x}$[14] can be found by taking only first order terms in Equation 2.4 and rearranging:

$$\tilde{\mathbf{x}}(t) - \mathbf{x}(t) = \delta\mathbf{x}(t) = d\mathbf{x}(t) - \dot{\mathbf{x}}(t)dt. \tag{2.5}$$

2.1.1.1 The State Transition Matrix

From Eqs. 2.1 and 2.2, the differential equation governing the variation $\delta\mathbf{x}$ can be written as

$$\dot{\delta\mathbf{x}}(t) = \dot{\tilde{\mathbf{x}}}(t) - \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t) + \delta\mathbf{x}(t)) - \mathbf{f}(\mathbf{x}(t)). \tag{2.6}$$

Employing a first order Taylor Series expansion gives a linear, time-varying differential equation of the form

$$\dot{\delta\mathbf{x}}(t) = A(t)\delta\mathbf{x}(t), \tag{2.7}$$

where $A(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\mathbf{x}(t)}$ is the Jacobian matrix evaluated along the current trajectory. Equation 2.7 has the well-known solution

$$\delta\mathbf{x}(t) = \Phi(t, t_0)\delta\mathbf{x}(t_0). \tag{2.8}$$

The matrix $\Phi(t, t_0)$ is the state transition matrix (STM), and it obeys the differential equation

$$\dot{\Phi}(t, t_0) = A(t)\Phi(t, t_0). \tag{2.9}$$

It is clear that the initial condition $\Phi(t_0, t_0)$ is the identity matrix $I_{n\times n}$, where n denotes the dimension of $\delta\mathbf{x}$, and that $\Phi(t, t_0)^{-1} = \Phi(t_0, t)$. Substituting the relationship defined in Equation 2.5 into Equation 2.8 yields

$$(d\mathbf{x}_f - \dot{\mathbf{x}}_f dt_f) = \Phi(t_f, t_0)\,(d\mathbf{x}_0 - \dot{\mathbf{x}}_0 dt_0) \tag{2.10}$$

The classical two-level targeting algorithm is based on the general variational equation described in Equation 2.10[25].

2.1.2 Classical Two-Level Targeting Algorithm

2.1.2.1 Level I Process

An impulsive two-level targeter[25] requires a startup arc represented by a series of N "patch states." These states, also termed "patch points," are selected by the user as representative waypoints along the trajectory. Consider a

dynamical system described by Equation 2.1, where $\mathbf{x}(t) = [\ \mathbf{r}(t)^T \quad \mathbf{v}(t)^T\ ]^T$. The user supplies the time and state at each patch point, t_{k-1} and $\mathbf{x}^+_{k-1} = [\ \mathbf{r}^T_{k-1} \quad \mathbf{v}^{+T}_{k-1}\ ]^T$ for $k = 2, \cdots, N+1$, respectively. Each state $\mathbf{x}^+_{k-1}$ is then numerically integrated forward over an interval $[t_{k-1}, t_k]$, for $k = 2, \cdots, N$. The integrated state, at time t_k, is recorded as $\mathbf{x}^-_k$. This is to allow for the possibility that the user supplied velocity at that point, $\mathbf{v}^+_k$, may not coincide with that identified during the propagation, $\mathbf{v}^-_k$. Such differences may arise due to a previously scheduled impulsive maneuver at that point or to differences in the models used (two- vs. three-body). If the state transition matrix $\Phi(t_k, t_{k-1})$ is partitioned into four 3×3 submatrices such that

$$\Phi(t_k, t_{k-1}) = \begin{bmatrix} A_{k,k-1} & B_{k,k-1} \\ C_{k,k-1} & D_{k,k-1} \end{bmatrix} \tag{2.11}$$

then the variational equations associated with the trajectory segment from $k-1$ to k can be expressed in vector form as

$$\begin{bmatrix} d\mathbf{r}_k - \mathbf{v}^-_k dt_k \\ d\mathbf{v}^-_k - \mathbf{a}^-_k dt_k \end{bmatrix} = \begin{bmatrix} A_{k,k-1} & B_{k,k-1} \\ C_{k,k-1} & D_{k,k-1} \end{bmatrix} \begin{bmatrix} d\mathbf{r}_{k-1} - \mathbf{v}^+_{k-1} dt_{k-1} \\ d\mathbf{v}^+_{k-1} - \mathbf{a}^+_{k-1} dt_{k-1} \end{bmatrix}. \tag{2.12}$$

Traditionally, the Level I process adjusts the outgoing velocity $\mathbf{v}^+_{k-1}$ (i.e., the initial velocity of a trajectory segment) at each patch point to ensure position continuity along the full trajectory. This is graphically illustrated in Fig. 2.2(a)-2.2(b). If the initial and final times along the segment are fixed, then the required change to the initial velocity has a unique solution given by

$$d\mathbf{v}^+_{k-1} = B^{-1}_{k,k-1} d\mathbf{r}_k. \tag{2.13}$$

However, if the terminal time t_k is allowed to vary, additional flexibility can be gained by utilizing t_k as a control variable. In this case, the desired adjustments

to the control parameters are found through the minimum norm solution to the first variational equation of Equation 2.12,

$$\begin{bmatrix} d\mathbf{v}_{k-1}^{+} \\ dt_k \end{bmatrix} = \begin{bmatrix} B_{k,k-1} & \mathbf{v}_k^- \end{bmatrix}^T \left(\begin{bmatrix} B_{k,k-1} & \mathbf{v}_k^- \end{bmatrix} \begin{bmatrix} B_{k,k-1} & \mathbf{v}_k^- \end{bmatrix}^T \right)^{-1} d\mathbf{r}_k. \tag{2.14}$$

Thus, a traditional Level I process leads to a trajectory that is continuous in position but, potentially, discontinuous in velocity at certain points. This is rectified by incorporating a Level II correction.

The Level I process can also be modified to meet terminal constraints, rather than a terminal position vector[58]. Consider a constraint vector $\mathbf{y}$ of dimension $m \times 1$ defined by the general nonlinear vector function

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, t). \tag{2.15}$$

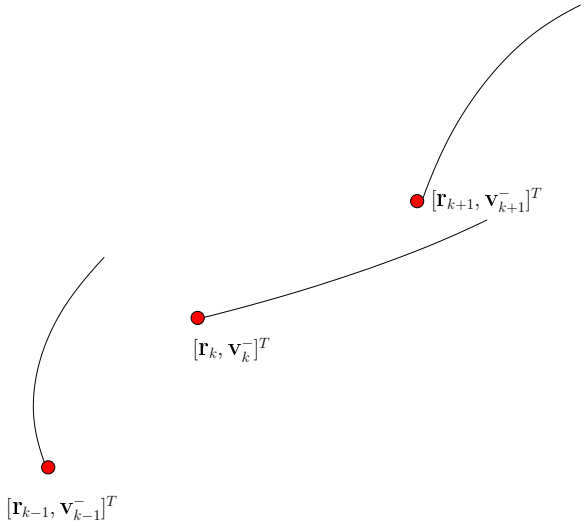
The linearized approximation of $\mathbf{y}$ at time t_k is

$$d\mathbf{y}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\Big|_{t_k,\mathbf{x}_k} \delta\mathbf{x}_k + \frac{d\mathbf{h}}{dt}\Big|_{t_k,\mathbf{x}_k} dt_k, \tag{2.16}$$

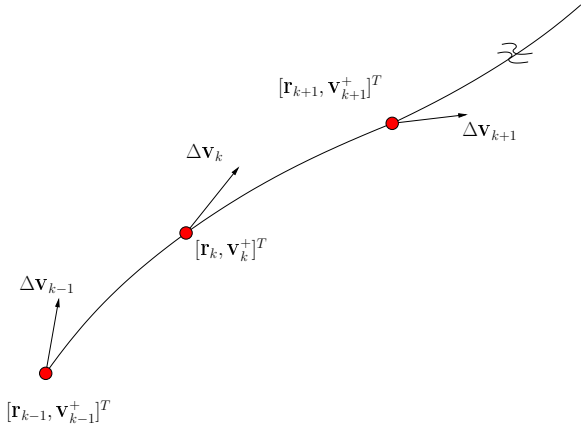
and $\mathbf{x}_k$ is related to $\mathbf{x}_{k-1}$ through the STM as previously discussed. Again using $\mathbf{v}_{k-1}^+$ and t_k as control parameters, the change in $\mathbf{y}_k$ as a function of the control parameters is given by

$$d\mathbf{y}_k = \left\{ \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\Big|_{t_k,\mathbf{x}_k} \begin{bmatrix} B_{k,k-1} & \mathbf{v}_k^- \\ D_{k,k-1} & \mathbf{a}_k^- \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{m\times 3} & \frac{d\mathbf{h}}{dt}\Big|_{t_k,\mathbf{x}_k} \end{bmatrix} \right\} \begin{bmatrix} d\mathbf{v}_{k-1}^{+} \\ dt_k \end{bmatrix}. \tag{2.17}$$

The desired control parameter adjustments are determined using the minimum norm solution, as before. This concept, targeting a constraint vector instead of a terminal state vector, is the foundation of the Level II correction procedure.



(a) Impulsive Two-Level Targeter: Before Level I Process is Applied



(b) Impulsive Two-Level Targeter: After Level I Process is Applied

Figure 2.2: Level I Process

2.1.2.2 Variable Scaling

Variable and constraint scaling is often used as a means of improving convergence in optimization routines[37], and this process could be similarly beneficial for linear targeting methods. In general, it is desirable for the partial derivatives of the constraints with respect to the control variables be O(1). This enables the targeter to evenly adjust control variables and meet constraints of varying orders of magnitude. For the Level I process, the scaled variables and partial derivatives would be

$$\mathbf{v}^+_{k-1_{scl}} = \frac{\mathbf{v}^+_{k-1}}{v_{scl}}, \quad t_{k_{scl}} = \frac{t_k}{t_{scl}}, \quad d\mathbf{r}_{k_{scl}} = \frac{d\mathbf{r}_k}{r_{scl}},$$

and

$$\frac{\partial \mathbf{r}_{k_{scl}}}{\partial \mathbf{v}^+_{k-1_{scl}}} = \frac{\partial \mathbf{r}_k}{\partial \mathbf{v}^+_{k-1}} \cdot \frac{v_{scl}}{r_{scl}}, \quad \frac{\partial \mathbf{r}_{k_{scl}}}{\partial t_{k_{scl}}} = \frac{\partial \mathbf{r}_k}{\partial t_k} \cdot \frac{t_{scl}}{r_{scl}}.$$

This modifies the equation for $d\mathbf{r}_k$ to

$$d\mathbf{r}_{k_{scl}} = \begin{bmatrix} \dfrac{\partial \mathbf{r}_{k_{scl}}}{\partial \mathbf{v}^+_{k-1_{scl}}} & \dfrac{\partial \mathbf{r}_{k_{scl}}}{\partial t_{k_{scl}}} \end{bmatrix} \begin{bmatrix} d\mathbf{v}^+_{k-1_{scl}} \\ dt_{k_{scl}} \end{bmatrix}. \tag{2.18}$$

The minimum norm solution to Equation 2.18 gives the scaled control parameter updates, which are then unscaled and implemented at the start of the next iteration. This procedure can be implemented in the Level II correction as well.

2.1.2.3 Level II Process

The Level II process adjusts the positions and times of each free patch state to drive any of the interior velocity discontinuities to zero, as well as meet

any additional user specified constraints, denoted here by the vector $\boldsymbol{\alpha}$. Figure 2.3(a)-2.3(c) shows this process graphically. Figure 2.3(a) is representative of the scenario in Fig. 2.2(b), where the trajectory is continuous in position but not in velocity. Figure 2.3(b) illustrates how the patch state positions, and potentially the associated times, have been adjusted by the Level II process. The system of equations that must be solved to find the control adjustments takes the general form

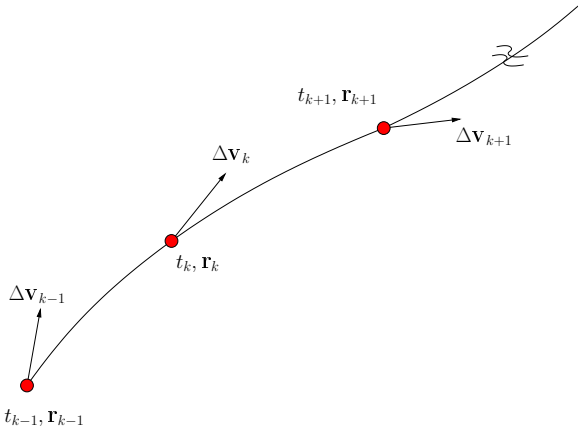
$$\left[\begin{array}{c} \Delta \mathbf{v} \\ \delta \boldsymbol{\alpha} \end{array} \right] = \left\{ \begin{array}{c} M \\ M_\alpha \end{array} \right\} \mathbf{b}, \tag{2.19}$$

where $\mathbf{b}$ is the vector of control parameters - patch state positions and times. The matrices M and M_α are composed of the partial derivatives of the velocity discontinuities (M) and the constraints (M_α) with respect to the corresponding control variables. Typically, a velocity discontinuity at patch point k is considered to be dependent only on the the patch state at k, the patch state immediately preceding it, and the patch state immediately following it ($k-1$ and $k+1$, respectively). Thus, the rows of M associated with $\Delta \mathbf{v}_k$ are given by

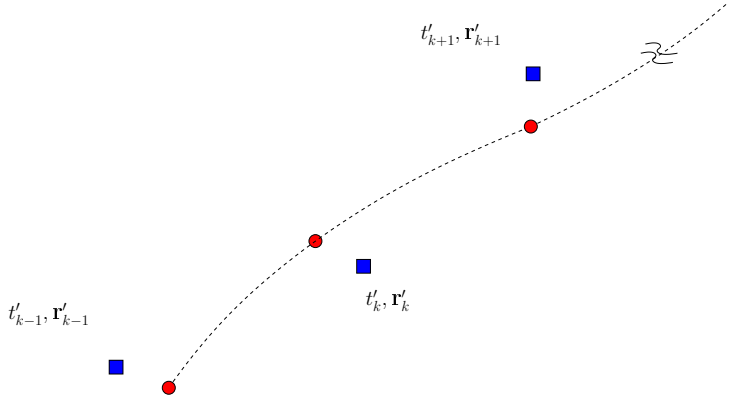
$$M_k = \left[\begin{array}{ccccccc} \mathbf{0}_{3\times 4(k-2)} & \frac{\partial \Delta \mathbf{v}_k}{\partial \mathbf{r}_{k-1}} & \frac{\partial \Delta \mathbf{v}_k}{\partial t_{k-1}} & \frac{\partial \Delta \mathbf{v}_k}{\partial \mathbf{r}_k} & \frac{\partial \Delta \mathbf{v}_k}{\partial t_k} & \frac{\partial \Delta \mathbf{v}_k}{\partial \mathbf{r}_{k+1}} & \frac{\partial \Delta \mathbf{v}_k}{\partial t_{k+1}} \quad \mathbf{0}_{3\times (N-4(k+1))} \end{array} \right]. \tag{2.20}$$

The elements of $\boldsymbol{\alpha}$ may depend on one or more patch states, including non-consecutive sets of patch states[58], depending on the nature of the constraint.

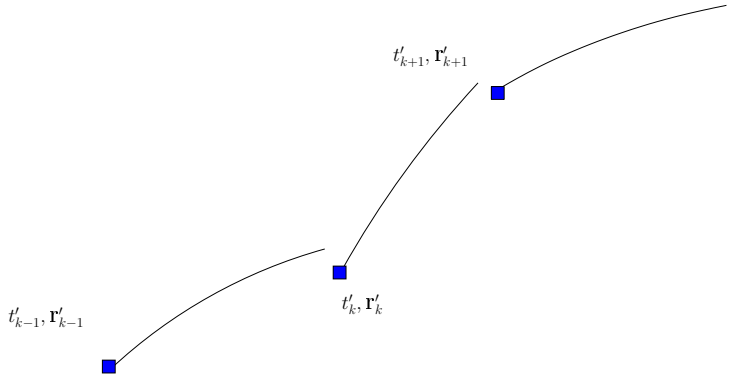
Since the corrections are linear in nature[25], propagation of the updated patch states in the nonlinear system can lead to a trajectory that is,



(a) Impulsive Two-Level Targeter: Before Level II Process is Applied



(b) Impulsive Two-Level Targeter: Corrections Suggested by Level II Minimum Norm Solution



(c) Impulsive Two-Level Targeter: After Propagating the Level II Updated Patch States

Figure 2.3: Level II Process

once again, discontinuous in position. The Level I process is subsequently applied again to generate an updated trajectory that is continuous in position. The combined Level I and Level II processes are generally repeated until the user specified tolerances are met for position and velocity continuity, as well as any additional constraints specified. Additional constraints may include velocity continuity at all patch states except where maneuvers are allowed, and interior or boundary constraints, among others[25].

It is important to note that the initial guess need not be feasible. That is, position/velocity/time continuity is not necessarily required for the targeter to successfully converge. However, since the overall process is based on linear systems theory, the initial discontinuities can impact the computation time. An initial guess with large discontinuities leads to an increased number of iterations. Naturally, an initial guess with absurdly large discontinuities can lead to non-convergence. Of course, a low quality initial guess can have a negative impact on both linear and nonlinear targeting algorithms. However, linear targeters will naturally be more sensitive to large errors. Developing a good initial guess is a problem within itself and highly dependent on the particular application of interest.

2.2 Preliminary Trajectory Design

Generating a suitable initial guess for trajectory design is frequently accomplished through the use of two-body analysis, such as a patched conic solution. The analytical expressions relating position and velocity at different

times along the orbit are, naturally, extremely useful for designing a trajectory to meet specified boundary conditions, although the resulting solutions often suffer from discontinuities or inaccuracies that can degrade the performance of a targeting algorithm. The methodology employed in this study also takes advantage of the analytical relationships of the two-body approximation, but incorporates them as part of an artificial potential function approach in an effort to mitigate some of these drawbacks.

2.2.1 The Two-Body Approximation

The n-body equations of motion for a satellite, relative to the primary body, are given by

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} - \sum_{j=2}^{n} \mu_j \left(\frac{\mathbf{r}_j - \mathbf{r}}{||\mathbf{r}_j - \mathbf{r}||^3} - \frac{\mathbf{r}_j}{r_j^3} \right) + \mathbf{a}_{pert} \tag{2.21}$$

where μ_j and $\mathbf{r}_j$ are the gravitational parameter and position of the jth body relative to the primary, respectively, and $\mathbf{a}_{pert}$ represents the sum of any perturbing accelerations (such as spherical harmonics, radiation pressure, etc.). Even without the perturbing accelerations, there is no analytical solution available in the n-body problem when $n \geq 3$. Thus, it is often advantageous to simplify the problem by neglecting everything except the first term in Equation 2.21, leaving the equations of motion as

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r}. \tag{2.22}$$

Equation 2.22 can be transformed through vector manipulations into equations which are perfect differentials, and the resulting constants of integration,

or integrals of the motion, describe the properties of the solution $\mathbf{r}(t)$[4]. The following section discusses several of these integrals, which are useful for determining transfer arcs between orbits.

2.2.1.1 Integrals of Motion

In a two-body model, ten integrals of the motion exist, six pertaining to linear momentum, three related to angular momentum, and lastly orbital energy. Of these, the most immediately useful quantities are angular momentum and energy. Taking the cross product of the position vector $\mathbf{r}$ with Equation 2.22 and integrating gives

$$\mathbf{r} \times \mathbf{v} = constant = \mathbf{h}, \tag{2.23}$$

where $\mathbf{h}$ is the specific angular momentum vector. By definition, this vector is always orthogonal to $\mathbf{r}$ and $\mathbf{v}$ and thus characterizes the orbital plane. Crossing Equation 2.22 with the constant vector $\mathbf{h}$ gives another perfect differential, whose solution can be written in the form

$$\mathbf{v} \times \mathbf{h} - \frac{\mu}{r}\mathbf{r} = constant = \mu\mathbf{e}. \tag{2.24}$$

The vector $\mathbf{e}$ is known as the eccentricity vector. Its magnitude, e, is equal to the eccentricity of the orbit and its direction indicates the orbit's periapsis.

Calculating $e^2 = \mathbf{e}^T\mathbf{e}$ from Equation 2.24 gives another constant of motion:

$$e^2 = 1 - \frac{h^2}{\mu}\left(\frac{2}{r} - \frac{v^2}{\mu}\right). \tag{2.25}$$

Because e and h are both constant, the quantity $\left(\frac{2}{r} - \frac{v^2}{\mu}\right)$ must be constant as well. The inverse of this quantity has units of length and is equal to the semi-major axis, a. If this relationship is expressed as

$$\frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a}, \tag{2.26}$$

the left-hand side of Equation 2.26 is immediately recognizable as the sum of the kinetic and potential energy of the orbit. Thus, the total energy of the orbit is constant and equal to the quantity $-\frac{\mu}{2a}$. An alternate form of Equation 2.26, which will be used for the remainder of this work, is the *vis-viva integral*:

$$v^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right). \tag{2.27}$$

The final constant of motion discussed here can be found by multiplying the transpose of both sides of Equation 2.24 with the position vector $\mathbf{r}$. After some rearranging, the resulting equation is

$$\frac{h^2}{\mu} = r + \mathbf{e}^T\mathbf{r}. \tag{2.28}$$

If ν is the angle between $\mathbf{e}$ and $\mathbf{r}$, also known as the true anomaly, then Equation 2.28 has the equivalent form

$$r = \frac{p}{1 + \mathbf{e}^T\hat{\mathbf{r}}} = \frac{p}{1 + e\cos\nu}, \tag{2.29}$$

where $p = \frac{h^2}{\mu}$ is the semi-latus rectum, or parameter. This is the polar equation of orbit.

2.2.2 Orbital Transfers

The integrals of motion described in the previous section can be used to calculate the necessary transfer velocity, $\mathbf{v}_t$, between two orbits in the two-body problem. Although much of the following analysis relates specifically to transfer between circular orbits, the equations in Subsection 2.2.1 apply to any conic orbit. The examination here of circular orbits is intended to provide insight into more complex general transfers that will be relevant in Chapter 4.

2.2.2.1 Coplanar Transfer

Transfer between two orbits can be accomplished with a single impulsive maneuver if and *only* if the orbits intersect. For coplanar orbits, this means that there exists a vector $\mathbf{r}_{int}$ with magnitude r_{int} and direction $\hat{\mathbf{r}}_{int}$ such that, from Equation 2.29,

$$r_{int} = \frac{p_0}{1 + \mathbf{e}_0^T \hat{\mathbf{r}}_{int}} = \frac{p_f}{1 + \mathbf{e}_f^T \hat{\mathbf{r}}_{int}}. \tag{2.30}$$

The subscripts 0 and f denote the initial and final orbits, respectively. Following some manipulation, Equation 2.30 can be expressed as

$$\frac{p_d - p_0}{||p_0\mathbf{e}_d - p_d\mathbf{e}_0||} = \frac{(p_0\mathbf{e}_d - p_d\mathbf{e}_0)^T}{||p_0\mathbf{e}_d - p_d\mathbf{e}_0||}\hat{\mathbf{r}}_{int} = \cos\psi, \tag{2.31}$$

where ψ is the angle between the vectors $(p_0\mathbf{e}_d - p_d\mathbf{e}_0)$ and $\hat{\mathbf{r}}_{int}$. Formulating the relationship in this manner gives a simple test for determining whether or not the two orbits intersect; because $|\cos\psi|$ cannot be greater than 1, an intersection point $\mathbf{r}_{int}$ can exist only if

$$\frac{|p_d - p_0|}{||p_0\mathbf{e}_d - p_d\mathbf{e}_0||} \leq 1. \tag{2.32}$$

If $\frac{|p_d - p_0|}{||p_0 \mathbf{e}_d - p_d \mathbf{e}_0||}$ is equal to 1, there will be exactly one intersection point. If it is less than 1, there are two points of intersection, and if the orbits coincide exactly, then the test term is undefined.

At least two maneuvers are required to transfer between orbits that do not intersect; the first maneuver places the vehicle on an intersection path with a point on the final orbit, while the second maneuver changes the vehicle's velocity to match the desired orbital velocity at the intersection point. In general, the fuel-optimal two-impulse transfer between any two orbits must satisfy the necessary condition[4]

$$\frac{\Delta \mathbf{v}_1}{\Delta \mathrm{v}_1} \cdot (\mathbf{v}_{\rho 1} - \mathbf{v}_{c1}) = \frac{\Delta \mathbf{v}_2}{\Delta \mathrm{v}_2} \cdot (\mathbf{v}_{\rho 2} - \mathbf{v}_{c2}) \,, \tag{2.33}$$

where the subscripts ρ and c denote the radial and chordal components of the terminal transfer velocities. Between two circular orbits, the optimum two-impulse transfer is the Hohmann transfer, shown in Figure 2.4. The Hohmann transfer arc is tangent to both the initial and final orbits and has a transfer angle of 180^o. If β is defined as the ratio of the final orbit radius to the initial orbit radius $\left(\beta \triangleq \frac{r_f}{r_0}\right)$, then relation of the total Δv for the Hohmann transfer to the initial circular speed v_0 is given by

$$\frac{\Delta \mathrm{v}_{tot}}{v_0} = \left(1 - \frac{1}{\beta}\right) \sqrt{\frac{2\beta}{1+\beta}} + \frac{1}{\sqrt{\beta}} - 1. \tag{2.34}$$

Equation 2.34 can be easily verified by applying Equation 2.27 at the initial and final points of the transfer arc. A Hohmann-type transfer - doubly cotangential, with transfer angle 180° - is also the optimum bi-impulsive transfer

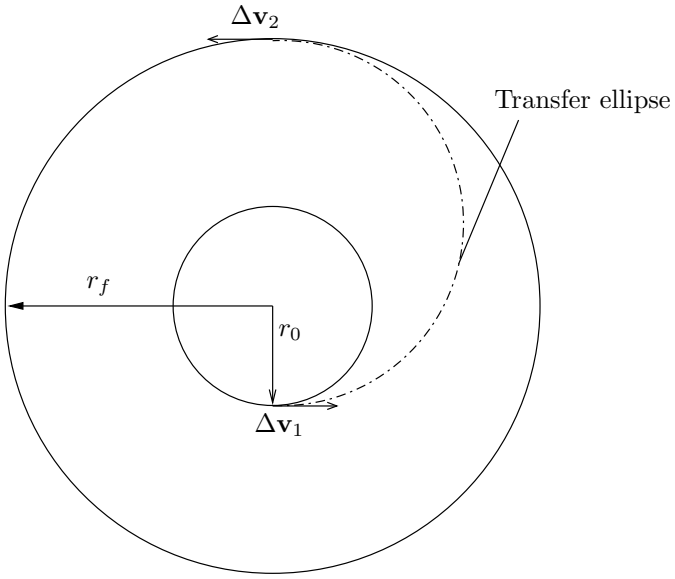


Figure 2.4: Coplanar Hohmann Transfer

between non-circular orbits if the apsidal lines of the two orbits are coincident and the initial and final transfer velocities are parallel.

Under certain circumstances for transfer between circular orbits, a reduction in total Δv can be achieved by adding a third maneuver such that the total transfer consists two 180° Hohmann-type transfer arcs, the first to an intermediate point $\mathbf{r}_t$ and second from $\mathbf{r}_t$ to the final orbit. This type of transfer, illustrated in Figure 2.5, is called a bi-elliptical transfer. Let $\alpha \triangleq \frac{r_t}{r_0}$ with β as defined previously. The ratio of the total Δv of the bi-elliptical transfer to the initial speed is then

$$\frac{\Delta \mathrm{v}_{tot}}{v_0} = \sqrt{\frac{2\alpha}{1+\alpha}} - 1 + \sqrt{\frac{2}{\alpha}}\left(\sqrt{\frac{\beta}{\beta+\alpha}} - \frac{1}{\sqrt{1+\alpha}}\right) + \frac{1}{\sqrt{\beta}}\left(\sqrt{\frac{2\alpha}{\alpha+\beta}} - 1\right). \tag{2.35}$$

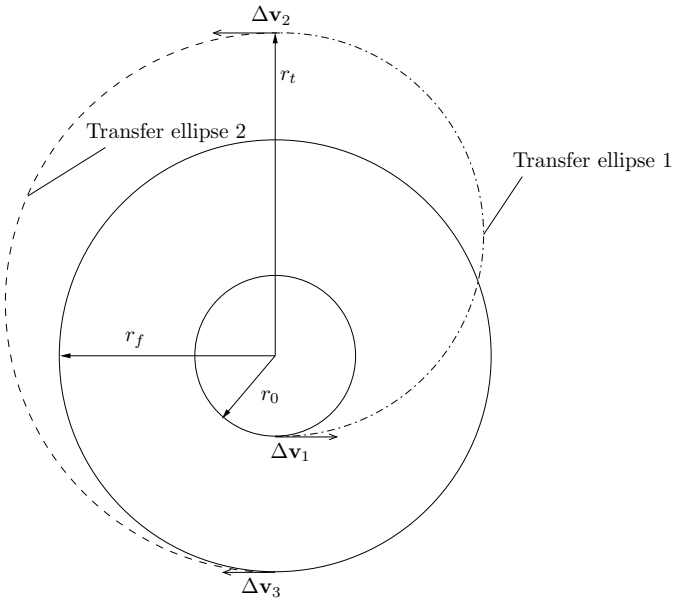


Figure 2.5: Coplanar Bi-elliptical Transfer

For sufficiently large β, there will always exist a value of α for which the bi-elliptical transfer has a lower total Δv than the Hohmann transfer. Figure 2.6 shows the characteristic velocities of a Hohmann transfer and several bi-elliptical transfers as a function of the ratio β $\left(\frac{r_f}{r_0}\right)$. From this figure, and from the closeup in Figure 2.7, it is evident that both β and α must be quite large to achieve any savings in Δv over the Hohmann transfer, thus requiring an exceptionally large transfer time, and even then the reduction is not substantial. For these reasons, the bi-elliptical transfer is rarely a practical option in the case of coplanar orbits.

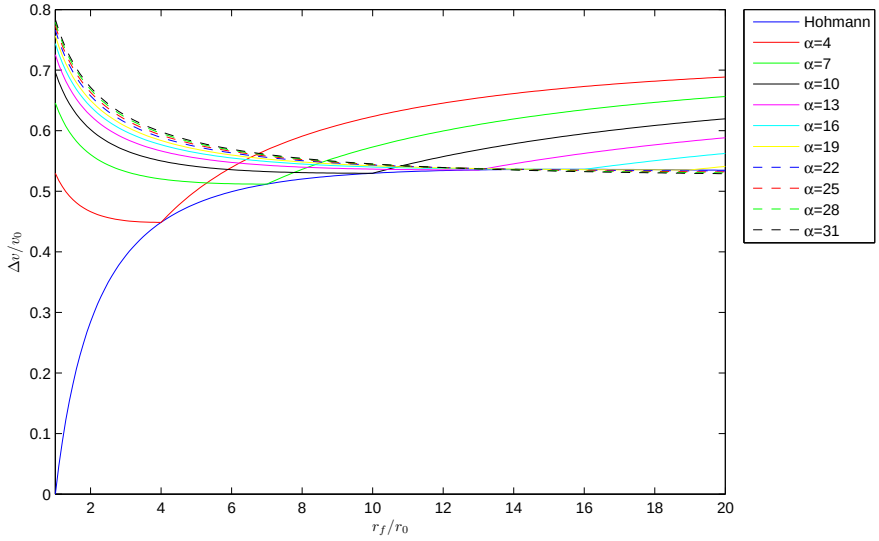


Figure 2.6: Hohmann vs. Bi-elliptical Coplanar Transfer

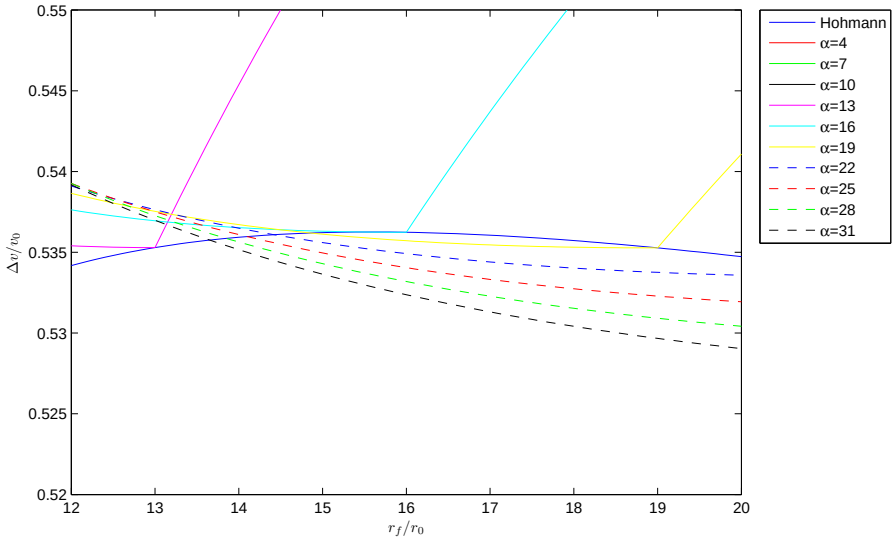


Figure 2.7: Hohmann vs. Bi-elliptical Coplanar Transfer, Closeup

2.2.2.2 Inclined Transfer

As stated previously, a single-impulse transfer between two orbits is only possible if the orbits intersect. In the case of non-coplanar orbits, the intersection point must lie along the intersection of the orbital planes, meaning that $\hat{\mathbf{r}}_{int}$ must be perpendicular to the angular momentum vectors of both the initial and the final orbit:

$$\hat{\mathbf{r}}_{int} = \pm \frac{\mathbf{h}_0 \times \mathbf{h}_f}{||\mathbf{h}_0 \times \mathbf{h}_f||}. \tag{2.36}$$

Given $\hat{\mathbf{r}}_{int}$, r_{int} can be computed for both orbits using Equation 2.29. Then if Equation 2.30 is satisfied, the orbits intersect at $\mathbf{r}_{int} = r_{int}\hat{\mathbf{r}}_{int}$.

As before, transfer between non-intersecting orbits can be achieved using two or more maneuvers. Versions of both the Hohmann transfer and the bi-elliptical transfer exist for inclined circular orbits, as shown in Figure 2.8. Now, however, the total Δv equation for each transfer must include the plane changes made at each maneuver. Keeping the same definitions for α and β, the Hohmann transfer Δv equation becomes

$$\frac{\Delta \mathrm{v}_{tot}}{v_0} = \left(1 + \frac{2\beta}{1+\beta} - 2\sqrt{\frac{2\beta}{1+\beta}}\cos\theta_1\right)^{1/2} + \frac{1}{\sqrt{\beta}}\left(1 + \frac{2}{1+\beta} - 2\sqrt{\frac{2}{1+\beta}}\cos\theta_2\right)^{1/2}, \tag{2.37}$$

and the Δv requirement for the bi-elliptical transfer is given by[3]

$$\begin{aligned}\frac{\Delta \mathrm{v}_{tot}}{v_0} = &\left(1+\frac{2\alpha}{1+\alpha}-2\sqrt{\frac{2\alpha}{1+\alpha}}\cos\theta_1\right)^{1/2} \\ &+\sqrt{\frac{2\beta}{\alpha(\alpha+\beta)}}\left(1+\frac{\alpha+\beta}{\beta(1+\alpha)}-2\sqrt{\frac{\alpha+\beta}{\beta(1+\alpha)}}\cos\theta_2\right)^{1/2} \\ &+\frac{1}{\sqrt{\beta}}\left(1+\frac{2\alpha}{\alpha+\beta}-2\sqrt{\frac{2\alpha}{\alpha+\beta}}\cos\theta_3\right)^{1/2},\end{aligned} \tag{2.38}$$

where θ_j is the plane change at the jth maneuver and the sum of the θ_js is equal to the total plane change θ_T. The optimal plane change split among the maneuvers minimizes $\frac{\Delta \mathrm{v}_{tot}}{v_0}$ subject to the constraint

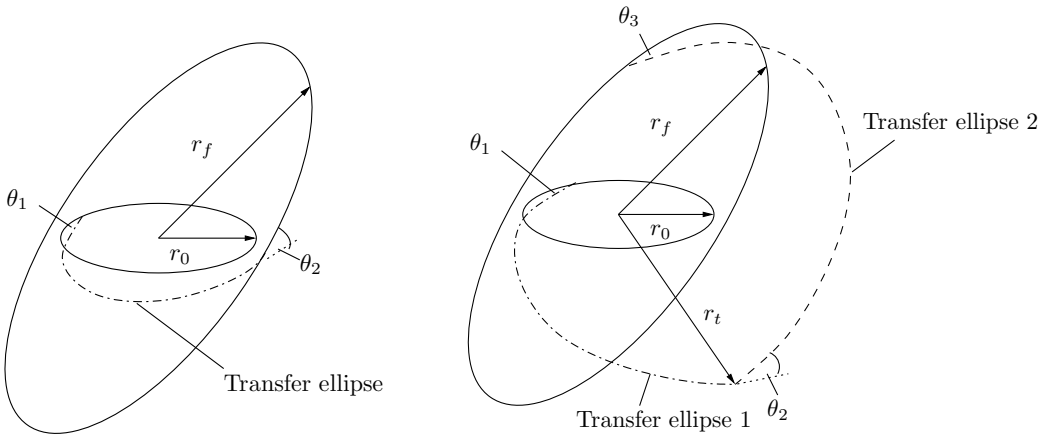
$$\theta_T - \sum_{j=1}^{N}\theta_j = 0, \tag{2.39}$$

where N is either 2 or 3, depending on whether a Hohmann transfer or a bi-elliptical transfer is used. The θ_j values are found by solving the set of equations

$$\frac{\partial}{\partial\theta_j}\left[\frac{\Delta \mathrm{v}_{tot}}{v_0}+\lambda\left(\theta_T-\sum_{j=1}^{N}\theta_j\right)\right]=0. \tag{2.40}$$

The scalar constant λ is a Lagrange multiplier.

Although no analytical solution is available, Equation 2.40 can be solved numerically to obtain the optimal plane change at each maneuver. Figure 2.9 shows the total Δv and optimal θ_1 and θ_2 for various values of α plotted against β, for a total plane change of 30°. It is evident that, for an inclined transfer, the bi-elliptical transfer is more economical than the Hohmann transfer at a lower value of β than for a coplanar transfer. Furthermore, when β

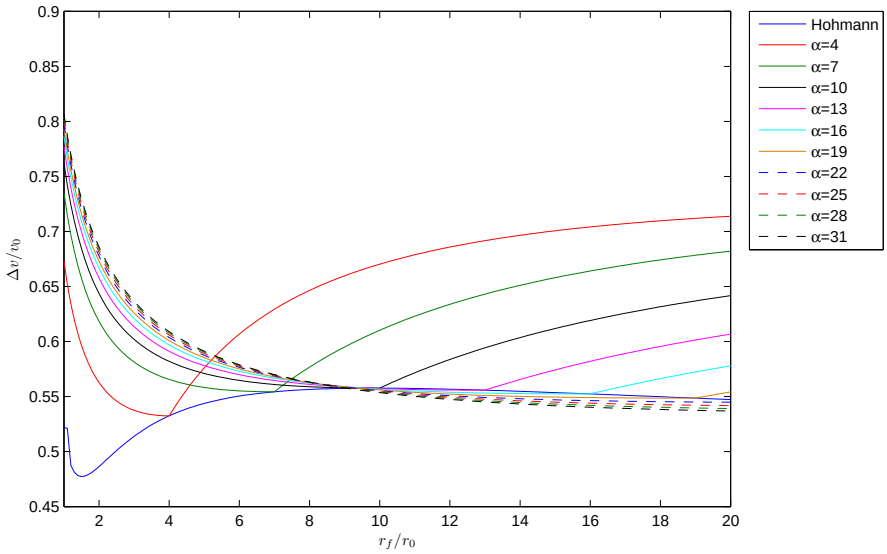


(a) Inclined Hohmann Transfer (b) Inclined Bi-elliptical Transfer

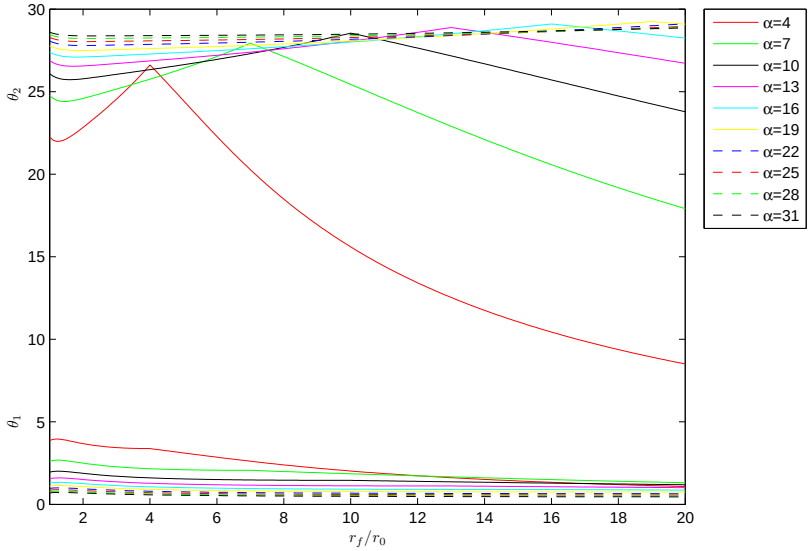
Figure 2.8: Inclined Hohmann and Bi-elliptical Transfers

is close to 1, a small reduction in Δv can be obtained by using a bi-elliptical transfer with α only slightly greater than β. This result is illustrated in Figure 2.10, again for a total plane change of 30°.

As the total plane change increases, the range of β values for which the bi-elliptical transfer requires a lower total Δv increases as well. This is demonstrated in Figure 2.11, which shows the Δvs and optimal θ_1 and θ_2 for a 50° total plane change. At this point, the two separate β ranges for which the bi-elliptical transfer is more fuel efficient (β near unity and β very high) have merged, so that there is always a bi-elliptical transfer that is more economical than the inclined Hohmann transfer. From experimentation, the critical plane change value at which these two regions merge appears to be around 41°.



(a) Δv



(b) Plane Change Split

Figure 2.9: Hohmann vs. Bi-elliptic Inclined Transfer, 30^o

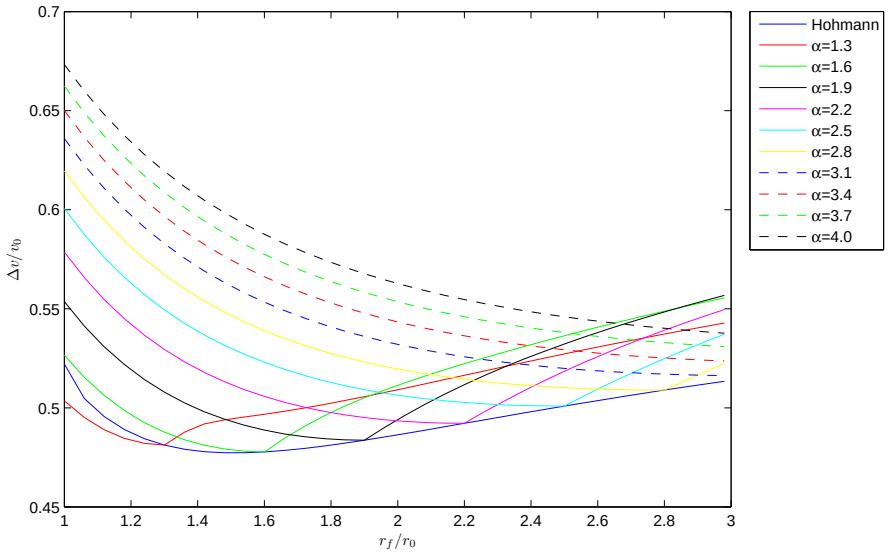


Figure 2.10: Hohmann vs. Bi-elliptical Inclined Transfer, β Near Unity

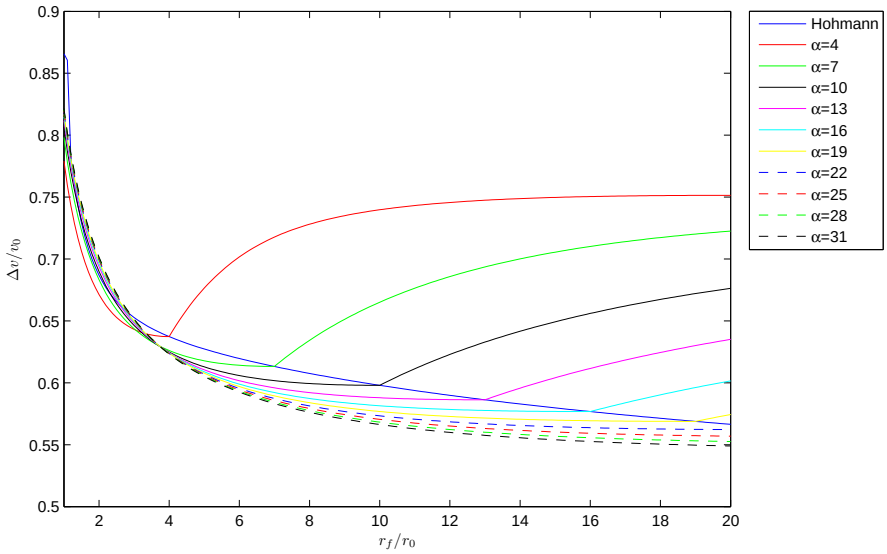
2.3 Artificial Potential Function Methods

2.3.1 General Approach

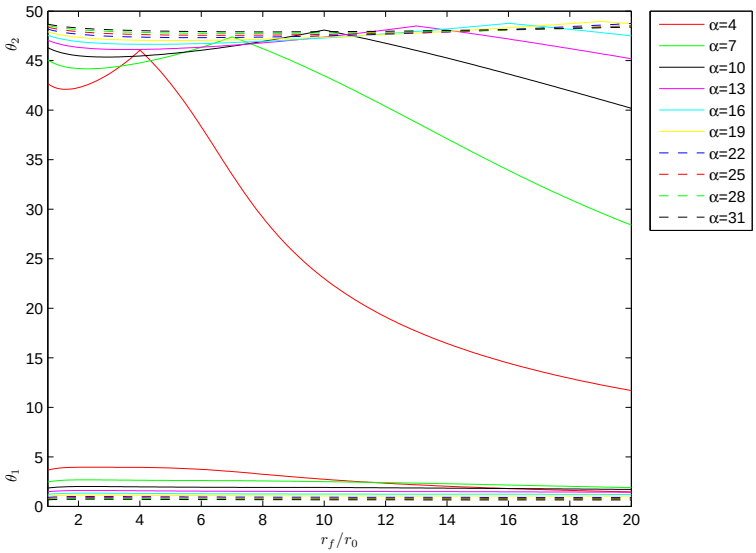
As stated in the previous chapter, artificial potential functions provide a means of mathematically representing the vehicle environment. The control acceleration of the vehicle is equal to the negative gradient of the potential:

$$\mathbf{a}_c = -\nabla P. \tag{2.41}$$

This drives the vehicle towards the minimum of the potential and away from any regions of high potential. If the potential field is constructed properly, this behavior corresponds to an approach towards the goal state while simultaneously avoiding any obstacles in the vicinity. The value of the potential field is typically written as a function of the distance between the vehicle and



(a) Δv



(b) Plane Change Split

Figure 2.11: Hohmann vs. Bi-elliptic Inclined Transfer, 50^o

Table 2.1: Potential Field Features

	x	y	κ
Goal	3	0	1
Obstacle 1	6	4	45
Obstacle 2	3	3	25
Obstacle 3	2	6	30

its surrounding features (goals or obstacles). For example, if the goal position is defined as $\mathbf{r}_g$ and various obstacles exist at $\mathbf{r}_{ob_j}$, $(j = 1, 2, ...)$, a simple potential may be defined as[51]

$$P = \kappa_g ||\mathbf{r}_g - \mathbf{r}||^2 + \sum_{j=1}^{N} \eta \kappa_j \left(\frac{1}{1 + ||\mathbf{r}_{ob_j} - \mathbf{r}||^2} - \frac{1}{\gamma r_{\eta_j}} \right), \tag{2.42}$$

where r_{η_j} is the radius of influence of the jth obstacle, γ is a user-defined constant, and

$$\eta = \begin{cases} 0 & \text{if } ||\mathbf{r}_{ob_j} - \mathbf{r}|| > r_{\eta_j} \\ 1 & \text{otherwise} \end{cases} \tag{2.43}$$

Figure 2.12 shows the potential field that results from Equation 2.42. The positions of the goal and the three obstacles, along with their respective κ values, are given in Table 2.1. The field as a whole slopes down toward its minimum at the goal position, with the exception of the three peaks centered at the obstacle positions.

Writing the potential as a function of only position, however, can result in undesirable oscillations about the goal point. Thus, a velocity term is often included to provide damping:

$$P = \kappa_g ||\mathbf{r}_g - \mathbf{r}||^2 + \sum_{j=1}^{N} \eta \kappa_j \left(\frac{1}{1 + ||\mathbf{r}_{ob_j} - \mathbf{r}||^2} - \frac{1}{\gamma r_{\eta_j}} \right) + \kappa_v \mathbf{v}^T \mathbf{v}. \tag{2.44}$$

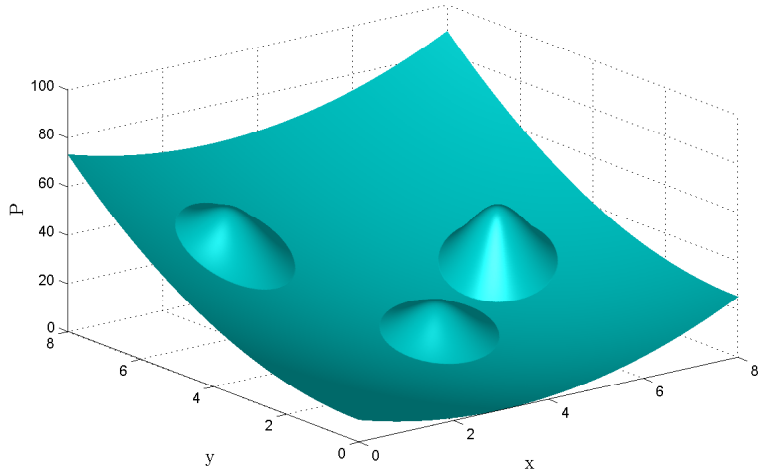


Figure 2.12: Example Potential Field

For $\kappa_v = 3$ and assuming an inertial environment (that is, $\ddot{\mathbf{r}} = \mathbf{a}_c$), the resulting vehicle trajectory is shown in Figure 2.13. With the velocity term adding damping, the path successfully avoids the obstacles (at the red X's) and terminates at the desired goal point without oscillating.

2.3.2 Velocity Error Approach

An alternative way to construct the potential function is by using the velocity error, rather than position error[31, 50]. In this approach, a desired velocity field is created based on the vehicle objectives. The potential is then defined as a function only of the difference between the current and desired velocities,

$$P = \kappa_v(\mathbf{v} - \mathbf{v}_d)^T(\mathbf{v} - \mathbf{v}_d). \tag{2.45}$$

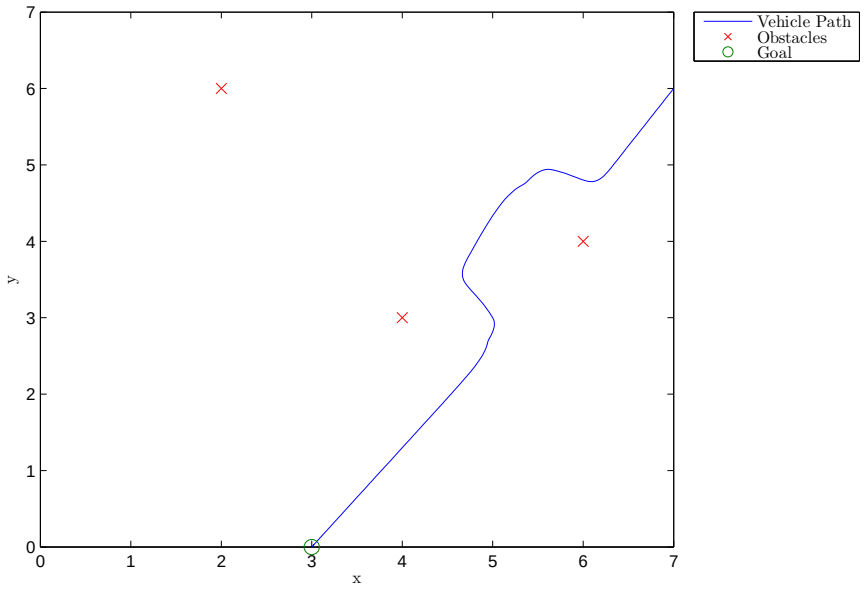


Figure 2.13: Potential Field Trajectory

If, as in the previous section, the goal of the vehicle is to arrive at a specified target position, then the desired velocity field is constructed as a function of the position error. Similar to the position-based potential function in Equation 2.42, the velocity field seeks to drive the vehicle both towards the goal and away from any obstacles. An example velocity field is given by

$$\mathbf{v}_d = \frac{\kappa_g}{1 + ||\mathbf{r}_g - \mathbf{r}||} (\mathbf{r}_g - \mathbf{r}) - \sum_{j=1}^{N} \frac{\eta \kappa_j}{1 + ||\mathbf{r}_{ob_j} - \mathbf{r}||} \left(\mathbf{r}_{ob_j} - \mathbf{r}\right), \tag{2.46}$$

with η retaining its definition from Equation 2.43. This formulation, along with the goal and obstacle positions listed in Table 2.1, produces the velocity field shown in Figure 2.14. The κ_j values in Table 2.1 have been scaled by 0.025. Taking the gradient of P and substituting this definition of $\mathbf{v}_d$ gives the control acceleration as

$$\mathbf{a}_c = -2\kappa_v \left[\mathbf{v} - \frac{\kappa_g}{1 + ||\mathbf{r}_g - \mathbf{r}||} (\mathbf{r}_g - \mathbf{r}) - \sum_{j=1}^{N} \frac{\eta \kappa_j}{1 + ||\mathbf{r}_{ob_j} - \mathbf{r}||} \left(\mathbf{r}_{ob_j} - \mathbf{r}\right)\right]^T. \tag{2.47}$$

This result is similar to the control acceleration obtained by taking the gradient of Equation 2.44 in that it includes a velocity term for damping along with the position error and obstacle proximity terms. The trajectory that results from the control in Equation 2.47 is displayed in Figure 2.15. As in the previous section, the vehicle successfully navigates the obstacles and terminates at the target position.

The velocity error approach is of particular interest as a trajectory design tool. Construction of an appropriate desired velocity field for a vehicle under gravitational influence can be accomplished in a fairly straightforward

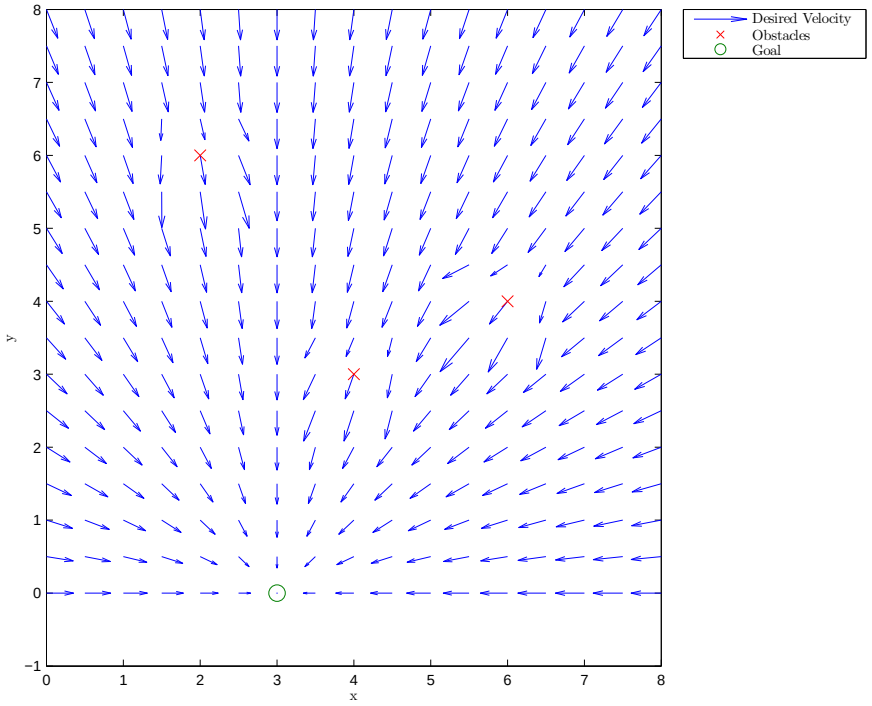


Figure 2.14: Velocity Field

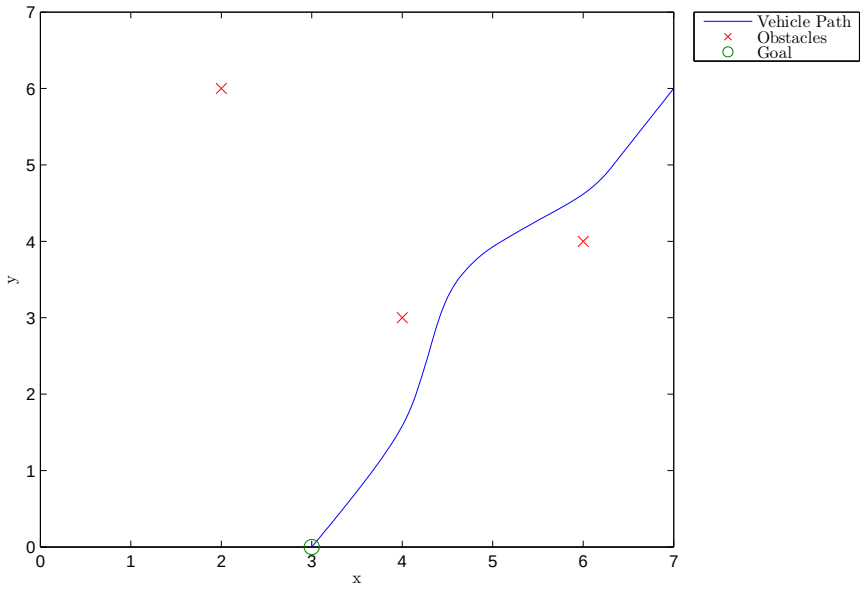


Figure 2.15: Potential Field Trajectory

manner using the relationships described in Section 2.2. Additionally, the minimum of the velocity error corresponds to the maneuver point requiring the least Δv for the current orbital transfer objective. This can help to alleviate the suboptimal nature of artificial potential function methods.

Chapter 3

Continuous Actuation Targeting Algorithm

The basic structure of a two-level targeter that incorporates continuous control is generally similar to that of the impulsive two-level targeter[25]. The two-level framework offers significant advantages in terms of computational efficiency, flexibility, and robustness[25]. The design of the continuous actuation targeting algorithm is intended to retain the advantages of the two-level structure while extending its range of application to include problems with finite periods of control actuation. Both algorithms utilize a linearized dynamical model and employ a minimum norm solution in computing the updates to the control variables. The differences, which subsequently lead to added complexity and computational overhead, stem from the increased dimensionality of the state vector associated with any controlled arc. Due to the interdependency between the state variables, the partial derivatives are also more complex in nature than those of the impulsive targeter.

Provided a suitable initial guess is available, the formulation of the impulsive two-level targeter[25] is generalized in nature. As such, it can be applied to any problem that employs impulsive corrections. However, problems that employ continuous control of any kind cannot benefit from this approach,

at least not in its original form. The key to transitioning the methodology to address problems that include finite control segments is to formulate the control variables in terms of initial parameters that can be adjusted. For example, in the case of finite burn targeting, if the thrust vector is inertially fixed and the engine only allows fixed thrust or acceleration levels, the control variables become the time of ignition, and the direction and duration of the burn. Under similar conditions, if linear steering is allowed, the control variables become the time of ignition, the duration of the burn, the initial burn direction, and the rate of change of the burn direction. This is discussed in greater detail in Section 3.1.3.

In the classical impulsive two-level targeter, the Level I process employs Δv's at the start of each segment to achieve position continuity. These Δv's, and – if desired – the time at which the maneuvers are executed, are control variables in that case. In a continuous actuation process, the Level I control variables include any parameters on which the control behavior depends, along with any on/off or switching times associated with the control input. The structure of the continuous actuation two-level targeter is subsequently developed and presented here.

3.1 Level I Process

As previously discussed, the application of a Level I process[25] typically involves the identification of an arc that spatially connects two points in space. For the orbital transfer problem, this is the n-body analog to a two-

body Lambert targeter, except the time of flight is not necessarily fixed or pre-specified. This ultimately reduces to a form of linear differential correction where Δv's are adjusted to meet the specified goals. In the present study, however, impulsive maneuvers are assumed to not adequately model the true nature of the control implementation. Thus, the Level I process traditionally employed in the two-level targeter[25] requires some modification to incorporate finite actuation periods. In a Level I process that employs continuous control rather than impulsive maneuvers, the controlled arc is treated as a subsegment of the total arc between patch points $k-1$ and k.

In the impulsive case, the time derivative $\dot{\mathbf{x}}(t)$ is a function only of the state $\mathbf{x}(t)$ and the time t, as shown in Equation 2.1. With the addition of continuously actuated subarcs, however, the contribution of the control acceleration must also be taken into account:

$$\dot{\mathbf{x}}(t) = \tilde{\mathbf{f}}(\mathbf{x}(t), t, \mathbf{u}(t, \boldsymbol{\zeta}(t))). \tag{3.1}$$

The control input is represented by $\mathbf{u}(t)$, and $\boldsymbol{\zeta}$ is the vector of adjustable parameters that directly influence the control acceleration. The elements of $\boldsymbol{\zeta}$ are not required to be constant, as long as the kinematics of $\boldsymbol{\zeta}$ can be expressed by

$$\dot{\boldsymbol{\zeta}}(t) = \mathbf{f}_{\zeta}(\boldsymbol{\zeta}(t), \mathbf{x}(t), t), \tag{3.2}$$

where the function $\mathbf{f}_{\zeta}$ is known.

For the system to remain consistent with Equation 2.1, it is necessary to consider an augmented state vector $\tilde{\mathbf{x}}_{k-1}^{+} = \begin{bmatrix} \mathbf{x}_{k-1}^{+T} & \boldsymbol{\zeta}_{k-1}^{T} \end{bmatrix}^{T}$. In determining

the Level I algorithm, the goal is to identify a relation between the target, which can be either the terminal state vector at point k or a terminal constraint vector, and the control variables. The general form for the Jacobian of the augmented state vector is

$$\frac{\partial \dot{\tilde{\mathbf{x}}}}{\partial \tilde{\mathbf{x}}} = \begin{bmatrix} \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{x}} & \frac{\partial \dot{\mathbf{x}}}{\partial \boldsymbol{\zeta}} \\ \frac{\partial \dot{\boldsymbol{\zeta}}}{\partial \mathbf{x}} & \frac{\partial \dot{\boldsymbol{\zeta}}}{\partial \boldsymbol{\zeta}} \end{bmatrix}. \tag{3.3}$$

The upper left block of the augmented Jacobian is equivalent to the Jacobian of the original state vector. From Equation 3.3, the associated state transition matrix $\Phi(t, t_0)$ can be propagated to find the variational equations for the augmented state:

$$\begin{bmatrix} d\mathbf{x}_f - \dot{\mathbf{x}}_f dt_f \\ d\boldsymbol{\zeta}_f - \dot{\boldsymbol{\zeta}}_f dt_f \end{bmatrix} = \Phi(t_f, t_0) \begin{bmatrix} d\mathbf{x}_0 - \dot{\mathbf{x}}_0 dt_0 \\ d\boldsymbol{\zeta}_0 - \dot{\boldsymbol{\zeta}}_0 dt_0 \end{bmatrix}. \tag{3.4}$$

This relates the terminal state to both the initial state and the initial control parameters.

3.1.1 Level I Targeting Example: Damped Nonlinear Pendulum

As a simple example to illustrate the concepts described above, consider a damped nonlinear pendulum with constant control input u in the positive θ direction, as shown in Figure 3.1. Assuming that the pendulum rod is massless and any friction is negligible, the dynamics of this system are described by

$$\ddot{\theta} = -\frac{g}{L}\sin\theta - \kappa\dot{\theta} + u, \tag{3.5}$$

where L is the length of the pendulum and κ is the damping constant. Using the traditional state vector for this problem, $\mathbf{x} = \begin{bmatrix} \theta & \dot{\theta} \end{bmatrix}^T$, the linearized

equations of motion are

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L}\cos\theta & -\kappa \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{x}_{current}} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{3.6}$$

where the notation $\mathbf{x}_{current}$ indicates that the Jacobian is evaluated along the current trajectory. As discussed previously, this form of the linearized equations is inconsistent with the necessary form for implementing the Level I process. If, however, the state vector is augmented to include the control, such that $\mathbf{x} = \begin{bmatrix} \theta & \dot{\theta} & u \end{bmatrix}^T$, the linearized equation becomes

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{g}{L}\cos\theta & -\kappa & 1 \\ 0 & 0 & 0 \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{x}_{current}} \begin{bmatrix} \theta \\ \dot{\theta} \\ u \end{bmatrix}. \tag{3.7}$$

Now, the linearized system equations are compatible with the Level I framework.

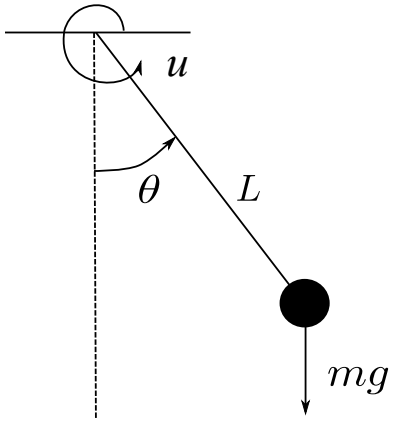


Figure 3.1: Pendulum with Control Input

It is assumed that the control input magnitude is constant but adjustable and that it can be turned off at some time t_T, so that $u = u_0 = \text{const}$

for $t_0 \leq t < t_T$ and $u = 0$ for $t_T \leq t \leq t_f$. Thus, the full trajectory segment can be split into two subsegments, one controlled and one uncontrolled, and the control parameters are the initial control magnitude u_0 and the cutoff time t_T. The variational equations for the uncontrolled subsegment can be written in the traditional form, since no augmentation of the state vector is necessary:

$$\begin{bmatrix} d\theta_f - \dot{\theta}_f^- dt_f \\ d\dot{\theta}_f - \ddot{\theta}_f^- dt_f \end{bmatrix} = \begin{bmatrix} a_{f,T} & b_{f,T} \\ c_{f,T} & d_{f,T} \end{bmatrix} \begin{bmatrix} d\theta_T - \dot{\theta}_T^+ dt_T \\ d\dot{\theta}_T - \ddot{\theta}_T^+ dt_T \end{bmatrix}. \tag{3.8}$$

For the controlled subsegment, the variational equations for the augmented state are

$$\begin{bmatrix} d\theta_T - \dot{\theta}_T^- dt_T \\ d\dot{\theta}_T - \ddot{\theta}_T^- dt_T \\ du_T - \dot{u}_T^- dt_T \end{bmatrix} = \begin{bmatrix} a_{T,0} & b_{T,0} & e_{T,0} \\ c_{T,0} & d_{T,0} & f_{T,0} \\ h_{T,0} & i_{T,0} & j_{T,0} \end{bmatrix} \begin{bmatrix} d\theta_0 - \dot{\theta}_0^+ dt_0 \\ d\dot{\theta}_0 - \ddot{\theta}_0^+ dt_0 \\ du_0 - \dot{u}_0^+ dt_0 \end{bmatrix}. \tag{3.9}$$

Since there are no impulsive control inputs, the trajectory for $\theta(t)$ is smooth and $\dot{\theta}_T^+ = \dot{\theta}_T^-$. Furthermore, it can be determined from Equation 3.5 that $\ddot{\theta}_T^+ = \ddot{\theta}_T^- - u$. Thus, the expressions for $d\theta_T - \dot{\theta}_T^- dt_T$ and $d\dot{\theta}_T - \ddot{\theta}_T^- dt_T$ from Equation 3.9 can be substituted into Equation 3.8 to determine the relationship between the state at time t_f and the control variables u_0 and t_T. The desired change to the control parameters is found to be

$$\begin{bmatrix} du_0 \\ dt_T \end{bmatrix} = \begin{bmatrix} (a_{f,T}e_{T,0} + b_{f,T}f_{T,0}) & b_{f,T}\Delta u_T \\ (c_{f,T}e_{T,0} + d_{f,T}f_{T,0}) & d_{f,T}\Delta u_T \end{bmatrix}^{-1} \begin{bmatrix} d\theta_f \\ d\dot{\theta}_f \end{bmatrix}, \tag{3.10}$$

where Δu_T is the difference between the incoming and outgoing control input at t_T (in this case, u_0). It is important to note that the number of constraints cannot exceed the number of control parameters. For this problem, if both θ and $\dot{\theta}$ at t_f have specified target values, then there must be at least two

control variables available. In the formulation above, these are assumed to be u_0 and t_T. However, additional control parameters may also be added by increasing the number of subsegments in the overall trajectory and allowing the switching times between each subsegment to function as controls. This is demonstrated in the third numerical example below.

3.1.1.1 Numerical Example 1: Convergence to Natural Equilibrium

In this example, the goal is for both θ and $\dot{\theta}$ to be equal to 0 at time $t_f = 20$ seconds. The control parameters, as discussed above, are u_0 and t_T, the control cutoff time. From an initial guess of $u_0 = 3$ and $t_T = 3$, and initial conditions $\theta_0 = \pi/3$ radians and $\dot{\theta}_0 = 0$, the correction process converges in 3 iterations. The values of the controls and the resulting final state error at each iteration are listed in Table 3.1. Figure 3.2 shows the time history of θ for the final iteration, from which it is clear that the system does meet the desired terminal state. The red circle marks θ at the cutoff time t_T. Close examination of Figure 3.2 reveals the physical implication of the converged control parameters; at t_T, θ is both equal to the target final value of 0 radians and it is at an extremum of its oscillation about the equilibrium indicated by u_0, so that $\dot{\theta}$ is also equal to 0. By turning off the control input at exactly these conditions, the targeting algorithm is able to achieve the desired terminal state.

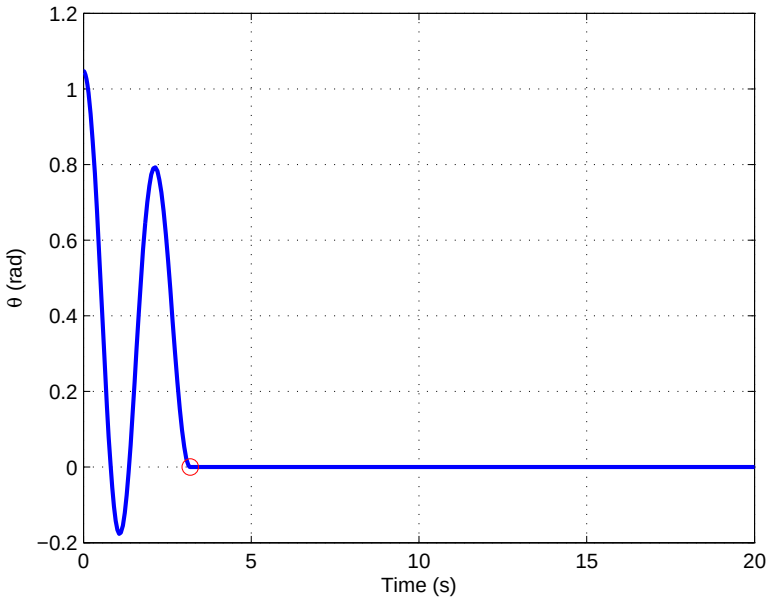


Figure 3.2: Pendulum θ Trajectory, Example 1

Table 3.1: Convergence Data, Pendulum Example 1

Iter.	u_0	t_T	θ Error (rad)	$\dot{\theta}$ Error (rad/s)
0	3.0000	3.0000	0.0016	-0.0190
1	3.0590	3.1930	-0.0016	-0.0015
2	3.3193	3.1799	0.0000	-0.0001
3	3.3187	3.1813	-0.0000	-0.0000

3.1.1.2 Numerical Example 2: Convergence to Forced Equilibrium

For this second example, it is desired to drive the pendulum to a non-natural final state, $\theta_d = \pi/6 = 0.5236$ rad. Because this is a forced equilibrium, it is no longer possible to simply turn off the control input prior to t_f, which eliminates t_T as a possible control parameter. With only one available control, u_0, there can be only one constraint, and so $\dot{\theta}_f$ must be left unrestricted. This will be addressed in the next example. In this case, with initial guess $u_0 = 1$, the algorithm again converges in 3 iterations, as shown in Table 3.2. As expected, the final iteration time history for θ in Figure 3.3 matches the specified θ_d at t_f but displays a nonzero $\dot{\theta}_f$. It should also be noted that the converged value of u_0 in Table 3.2 is slightly different than the analytical solution obtained by setting u_0 equal to $\frac{g}{L}\sin\theta_d$; this is due to the specified time at which θ_f must equal θ_d. Because t_f occurs before the oscillations have died out (as evidenced by Figure 3.3), there is no requirement that θ_f fall precisely at the equilibrium forced by u_0.

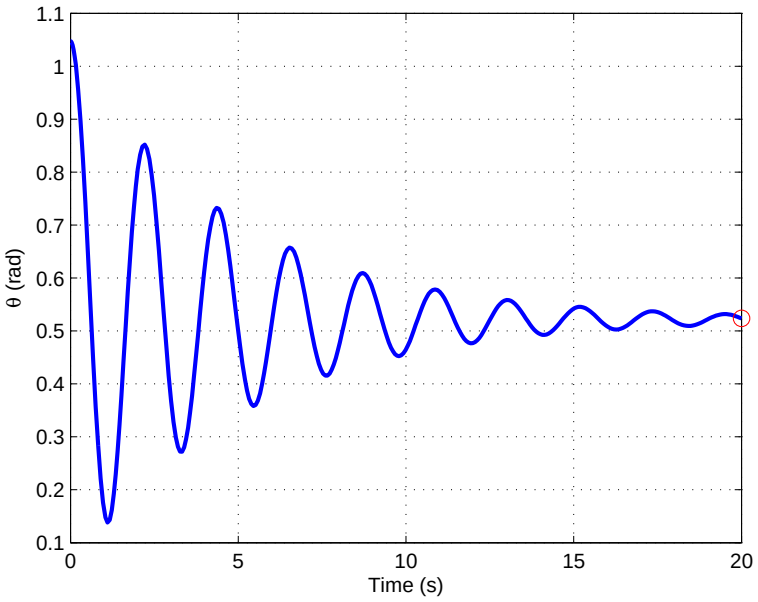


Figure 3.3: Pendulum θ Trajectory, Example 2

Table 3.2: Convergence Data, Pendulum Example 2

Iter.	u_0	θ Error (rad)
0	1.0000	0.4122
1	5.2068	-0.0422
2	4.8837	0.0008
3	4.8895	0.0000

3.1.1.3 Numerical Example 3: Step Control

In the previous example, the pendulum trajectory was treated as a single arc. The same problem is investigated here, except the trajectory is instead divided into two subarcs in order to increase the number of available control parameters. The control u is now a 2-segment step control scheme such that

$$u = \begin{cases} u_0 & \text{for } t_0 \leq t < t_T \\ u_T & \text{for } t_T < t \leq t_f \end{cases} \tag{3.11}$$

If u_T is assumed to be fixed at $u_T = \frac{g}{L}\sin\theta_d$ (thus enforcing an equilibrium point at θ_d), this still allows for two control parameters, u_0 and t_T, as before. Thus, both θ and $\dot{\theta}$ can be constrained at t_f. The control adjustments from Equation 3.10 still apply, except that now $\Delta u_T = u_0 - u_T$. The initial conditions (Iteration 0) and converged solution (Iteration 6) are given in Table 3.3. As in the first example, the physical significance of these final control values can be inferred from the plot of θ in Figure 3.4. The red circle again indicates θ at time t_T; at this point, the pendulum has reached the bottom of its oscillation about the equilibrium enforced by u_0, and $\dot{\theta}$ is 0. The algorithm adjusts the values of u_0 and t_T so that this occurs precisely at $\theta = \theta_d$, and

so the switch to $u = u_T$ simply holds the pendulum at the new equilibrium instead of causing any further oscillations.

Table 3.3: Convergence Data, Pendulum Example 3

Iter.	u_0	t_T	θ Error (rad)	$\dot{\theta}$ Error (rad/s)
0	7.0000	5.0000	-0.0155	-0.0173
6	5.8925	5.6746	0.0000	0.0000

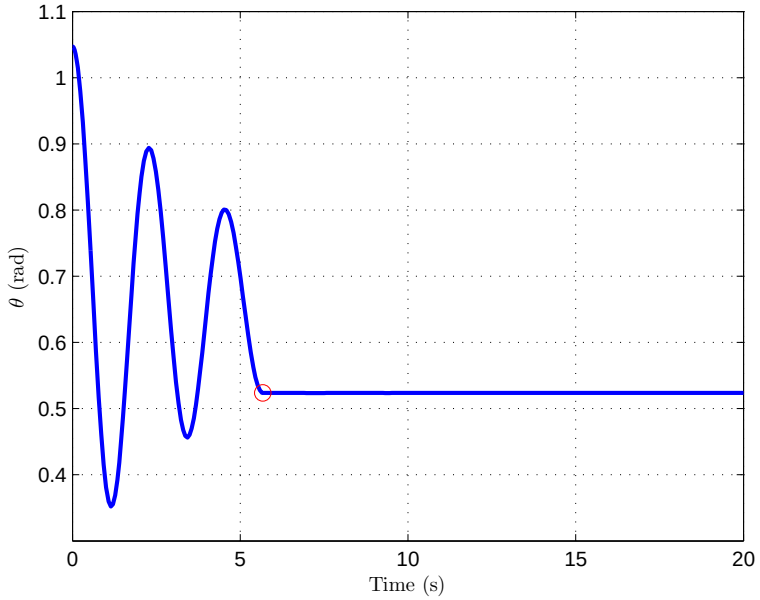


Figure 3.4: Pendulum θ Trajectory, Example 3

3.1.2 Level I Targeting Example: Attitude Slew Maneuver

Although the overall goal of this work is the development of an autonomous guidance process for orbital transfer, the targeting algorithm presented in this chapter is generalized in nature and can be applied to a variety

of different applications. To demonstrate the versatility of the algorithm, an attitude slew maneuver is investigated. The attitude representation for this problem is the quaternion, which is written as $\mathbf{q} \equiv \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T$ and must satisfy the constraint $\mathbf{q}^T\mathbf{q} = 1$. The vector portion of the quaternion is defined as $\mathbf{q}_v = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T = \hat{\mathbf{e}}\sin(\nu/2)$, where $\hat{\mathbf{e}}$ is the Euler axis of rotation and ν is the rotation angle. The scalar component is expressed as $q_4 = \cos(\nu/2)$[45]. The quaternion kinematic equations are

$$\dot{\mathbf{q}}_v = \frac{1}{2}\left(q_4 I_{3\times 3} + [\mathbf{q}_v\times]\right)\boldsymbol{\omega} \tag{3.12}$$

$$\dot{q}_4 = -\frac{1}{2}\mathbf{q}_v^T\boldsymbol{\omega}, \tag{3.13}$$

where $\boldsymbol{\omega}$ is the angular velocity vector in the vehicle body frame. Instead of a constant control input, as in the previous example, the control actuation for this system is provided by a set of 3 orthogonally-mounted reaction wheels, aligned with the body axes of the vehicle. Reaction wheels control the attitude by transferring momentum between the wheels and the vehicle, rather than by applying an external torque. This means that the vehicle can no longer be considered a rigid body. The total angular momentum of the system is the sum of $J\boldsymbol{\omega}$ and $\mathbf{h}_w$, and so the dynamics of the wheels must be included as an extra term in Euler's equations[17],

$$\dot{\boldsymbol{\omega}} = J^{-1}\left[-\boldsymbol{\omega}\times J\boldsymbol{\omega} - (\boldsymbol{\omega}\times\mathbf{h}_w + \mathbf{N}_w)\right], \tag{3.14}$$

where J is the vehicle inertia matrix, $\mathbf{h}_w$ is the angular momentum of the wheels, and $\mathbf{N}_w$ is the torque applied to the wheels. The equation of motion

for the reaction wheels is given by

$$\dot{\mathbf{h}}_w = \mathbf{N}_w, \tag{3.15}$$

The augmented state vector and associated linearized equations may be expressed as

$$\begin{bmatrix} \dot{\mathbf{q}}_v \\ \dot{q}_4 \\ \dot{\boldsymbol{\omega}} \\ \dot{\mathbf{h}}_w \\ \dot{\mathbf{N}}_w \end{bmatrix} = A(t) \begin{bmatrix} \mathbf{q}_v \\ q_4 \\ \boldsymbol{\omega} \\ \mathbf{h}_w \\ \mathbf{N}_w \end{bmatrix}, \tag{3.16}$$

The Jacobian $A(t)$ is given by

$$A(t) = \begin{bmatrix} -\frac{1}{2}[\boldsymbol{\omega}\times] & \frac{1}{2}\boldsymbol{\omega} & \frac{1}{2}\left(q_4 I_{3\times 3} + [\mathbf{q}_v\times]\right) & \mathbf{0} & \mathbf{0} \\ -\frac{1}{2}\boldsymbol{\omega}^T & \mathbf{0} & -\frac{1}{2}\mathbf{q}_v^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -J^{-1}\left\{[\boldsymbol{\omega}\times]J - J[\boldsymbol{\omega}\times] - [\mathbf{h}_w\times]\right\} & -J^{-1}[\boldsymbol{\omega}\times] & -J^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I_{3\times 3} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}\Bigg|_{\mathbf{x}=\mathbf{x}_{current}}, \tag{3.17}$$

where $[\boldsymbol{\sigma}\times]$ represents the cross product matrix for any vector $\boldsymbol{\sigma}$. The partitioning of the state transition matrix is as follows:

$$\Phi_{t_f,t_0} = \begin{bmatrix} A_{t_f,t_0} & B_{t_f,t_0} & C_{t_f,t_0} & D_{t_f,t_0} & E_{t_f,t_0} \\ F_{t_f,t_0} & G_{t_f,t_0} & H_{t_f,t_0} & K_{t_f,t_0} & L_{t_f,t_0} \\ M_{t_f,t_0} & N_{t_f,t_0} & O_{t_f,t_0} & P_{t_f,t_0} & Q_{t_f,t_0} \\ R_{t_f,t_0} & S_{t_f,t_0} & T_{t_f,t_0} & U_{t_f,t_0} & V_{t_f,t_0} \\ W_{t_f,t_0} & X_{t_f,t_0} & Y_{t_f,t_0} & Z_{t_f,t_0} & \Gamma_{t_f,t_0} \end{bmatrix}. \tag{3.18}$$

As before, let t_T denote a control input switching time, dividing the trajectory into two subsegments. The controls for this problem are $\mathbf{N}_{w0}$ and $\mathbf{N}_{wT}$, which are the applied wheel torque vectors at t_0 and t_T, and the switching

time. This gives a total of 7 available control parameters. The relationship between the final targeted states and the controls is

$$\begin{bmatrix} d\mathbf{q}_{vf} \\ dq_{4f} \\ d\boldsymbol{\omega}_f \end{bmatrix} = \begin{bmatrix} (A_{f,T}E_{T,0} + B_{f,T}L_{T,0} + C_{f,T}Q_{T,0} + D_{f,T}V_{T,0})\ E_{f,T} & \left\{C_{f,T}\left(\dot{\boldsymbol{\omega}}_T^- - \dot{\boldsymbol{\omega}}_T^+\right) + D_{f,T}\left(\dot{\mathbf{h}}_{wT}^- - \dot{\mathbf{h}}_{wT}^+\right)\right\} \\ (F_{f,T}E_{T,0} + G_{f,T}L_{T,0} + H_{f,T}Q_{T,0} + K_{f,T}V_{T,0})\ L_{f,T} & \left\{H_{f,T}\left(\dot{\boldsymbol{\omega}}_T^- - \dot{\boldsymbol{\omega}}_T^+\right) + K_{f,T}\left(\dot{\mathbf{h}}_{wT}^- - \dot{\mathbf{h}}_{wT}^+\right)\right\} \\ (M_{f,T}E_{T,0} + N_{f,T}L_{T,0} + O_{f,T}Q_{T,0} + P_{f,T}V_{T,0})\ Q_{f,T} & \left\{O_{f,T}\left(\dot{\boldsymbol{\omega}}_T^- - \dot{\boldsymbol{\omega}}_T^+\right) + P_{f,T}\left(\dot{\mathbf{h}}_{wT}^- - \dot{\mathbf{h}}_{wT}^+\right)\right\} \end{bmatrix} \times \begin{bmatrix} d\mathbf{N}_{w0} \\ d\mathbf{N}_{wT} \\ dt_T \end{bmatrix}. \tag{3.19}$$

However, in this problem, simply knowing how $d\mathbf{q}_{vf}$ relates to the control parameters is not sufficient for finding the necessary control adjustments to target the final attitude. Quaternion errors are calculated as multiplicative, rather than additive, errors; if the subscript d denotes the desired quaternion, the quaternion error $\mathbf{q}_{err}$ is written as

$$\mathbf{q}_{v_{err}} = \begin{bmatrix} q_{4_d}I_{3\times 3} + [\mathbf{q}_{v_d}\times] \\ -\mathbf{q}_{v_d}^T \end{bmatrix}^T \mathbf{q} \tag{3.20}$$

$$q_{4_{err}} = \mathbf{q}_d^T\mathbf{q}. \tag{3.21}$$

To correct a terminal quaternion error, it is necessary to employ the Level I constraint targeting described in the previous chapter. Because q_4 is, by definition, entirely dependent on the vector portion $\mathbf{q}_v$, it is only necessary to drive $\mathbf{q}_{v_{err}}$ to $\mathbf{0}$ to achieve the desired attitude. The terminal constraint equation is therefore written as

$$\mathbf{y}_q = \mathbf{q}_{v_{err}}, \tag{3.22}$$

and the total error vector is given by $\left[\begin{array}{cc} \mathbf{y}_q & d\boldsymbol{\omega} \end{array} \right]^T$. The partial derivative of the constraint vector $\mathbf{y}_q$ with respect to a general control parameter β is found by taking the indirect partials,

$$\frac{\partial \mathbf{y}_q}{\partial \beta} = \frac{\partial \mathbf{y}_q}{\partial \mathbf{q}_{vf}} \frac{\partial \mathbf{q}_{vf}}{\partial \beta} + \frac{\partial \mathbf{y}_q}{\partial q_{4f}} \frac{\partial q_{4f}}{\partial \beta} \tag{3.23}$$

where, from Eqs. 3.20-3.22,

$$\frac{\partial \mathbf{y}_q}{\partial \mathbf{q}_{vf}} = q_{4_d} I_{3\times 3} - [\mathbf{q}_{v_d} \times] \tag{3.24}$$

and

$$\frac{\partial \mathbf{y}_q}{\partial q_{4f}} = -\mathbf{q}_{v_d}. \tag{3.25}$$

Combining the equations above with the terminal state dependencies in Equation 3.19 gives a relationship between the terminal constraints and the controls, which can then be used to determine the desired changes to control variables.

3.1.2.1 Numerical Example 1: Slew from Rest

This example simulates a slew maneuver from rest at the initial attitude through a rotation of approximately 0.7679 radians (44°) to the specified terminal attitude. In all of the following numerical examples, the simulated spacecraft has inertia matrix

$$J = \left[\begin{array}{ccc} 158.0 & 6.0 & 12.0 \\ 6.0 & 154.0 & 3.5 \\ 12.0 & 3.5 & 151.0 \end{array} \right] \tag{3.26}$$

The target angular velocity at time $t_f = 50$ seconds is $\mathbf{0}$ rad/s. The initial and final values of the control parameters are listed in Table 3.4, and the resulting error quaternion and angular velocity trajectories are shown in Figure 3.5.

Note that in Figure 3.5(b), the angular rate appears to evolve linearly. This is due to a simplification in the dynamics that arises with these particular initial conditions, specifically $\boldsymbol{\omega} = \mathbf{0}$ and $\mathbf{h}_w = \mathbf{0}$. Because the total angular momentum of the system ($J\boldsymbol{\omega} + \mathbf{h}_w$) is constant and initially $\mathbf{0}$, it remains at $\mathbf{0}$ for all t. Therefore, the first two terms in Equation 3.14 go to zero, leaving

$$\dot{\boldsymbol{\omega}} = -J^{-1}\mathbf{N}_w. \tag{3.27}$$

Since N_w is constant over the time periods t_0 to t_T and t_T to t_f, this indicates that $\boldsymbol{\omega}(t)$ is a piecewise linear function of time given by

$$\boldsymbol{\omega}(t) = \begin{cases} -J^{-1}\mathbf{N}_{w0}(t - t_0) & \text{for } t_0 \leq t < t_T \\ \boldsymbol{\omega}(t_T) - J^{-1}\mathbf{N}_{wT}(t - t_T) & \text{for } t_T < t \leq t_f \end{cases} \tag{3.28}$$

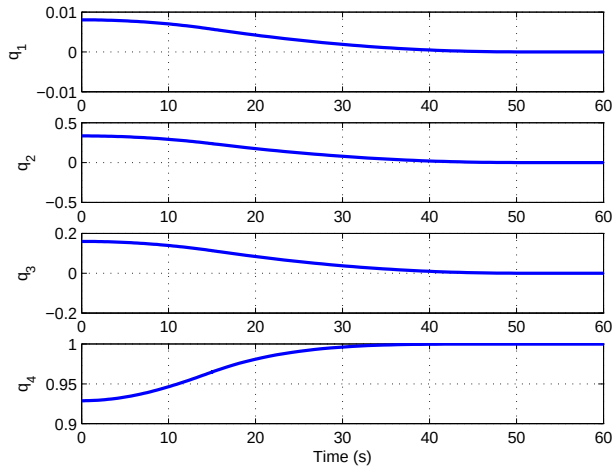
Additionally, it can be seen that the converged values of $\mathbf{N}_{w0}$, $\mathbf{N}_{wT}$, and t_T relate to one another such that $-J^{-1}\mathbf{N}_{w0}(t_T - t_0) = +J^{-1}\mathbf{N}_{wT}(t_f - t_T)$, thus zeroing out the angular velocity induced by $\mathbf{N}_{w0}$ so that the desired terminal value for $\boldsymbol{\omega}$ ($\mathbf{0}$ rad/s) is met.

Table 3.4: Convergence Data

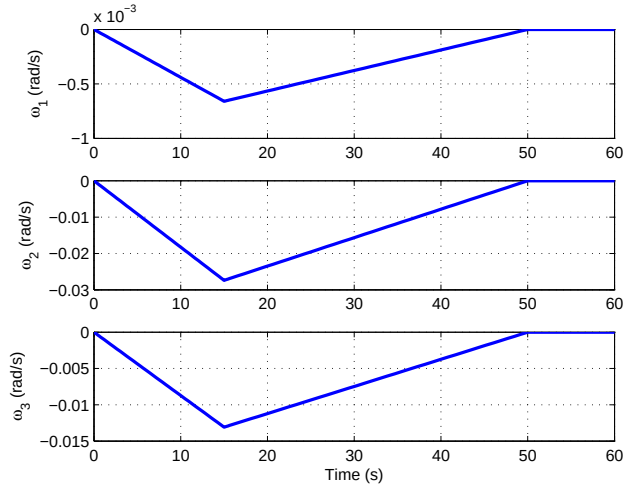
Iter.	$\mathbf{N}_{w0}^T$ (Nm)	$\mathbf{N}_{wT}^T$ (Nm)	t_T (s)	$\mathbf{q}_v$ Error
0	$\begin{bmatrix} 0.1000 & 0.1000 & 0.1000 \end{bmatrix}$	$\begin{bmatrix} -0.1000 & -0.1000 & -0.1000 \end{bmatrix}$	15.0	$\begin{bmatrix} 0.0002 & 0.3261 & 0.1549 \end{bmatrix}$
4	$\begin{bmatrix} 0.0284 & 0.2846 & 0.1386 \end{bmatrix}$	$\begin{bmatrix} -0.0122 & -0.1220 & -0.0594 \end{bmatrix}$	15.0015	$\begin{bmatrix} -0.0000 & -0.0000 & -0.0000 \end{bmatrix}$

3.1.2.2 Numerical Example 2: Slew from Slow Initial Rotation

In the second example, the vehicle is given a small initial angular rate of 0.0471 rad/s , or 2.7°/s. From the convergence data listed in Table 3.5, it is evident that even this minor initial rotation has a noticeable impact on the



(a) Attitude



(b) Angular Velocity

Figure 3.5: Attitude and Angular Velocity, Example 1

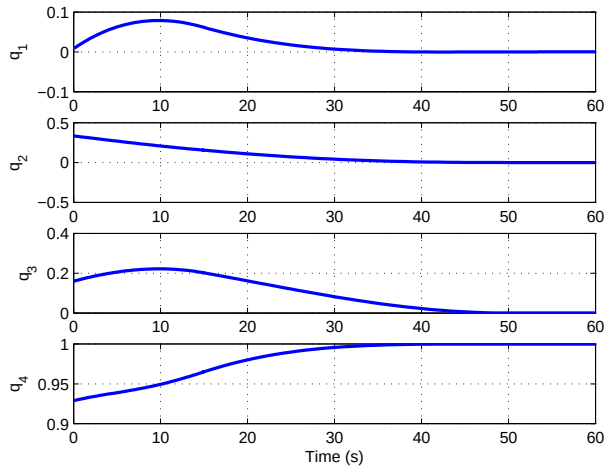
targeter results. The converged value for $\mathbf{N}_{w0}$ is significantly higher here than in the previous example, in order to counteract the undesired starting angular velocity. The algorithm also requires twice as many iterations to converge, indicating that this is a more straining case than a slew from rest. Figure 3.6 shows the effects of the initial rotation on the converged error quaternion (Fig. 3.6(a)) and angular velocity (Fig. 3.6(b)) trajectories. The first and third elements of the quaternion error initially deviate away from the target values, illustrating that the vehicle is initially rotating away from the desired final attitude; this is supported by the time history for $\boldsymbol{\omega}$, which shows the first and third elements crossing 0 (i.e. changing direction due to $\mathbf{N}_{w0}$) prior to t_T. In addition, the nonzero $\boldsymbol{\omega}_0$, combined with an initial wheel angular momentum $\mathbf{h}_w$ of $\mathbf{0}$, means that the cross terms in Equation 3.14 no longer cancel out, thus heightening the nonlinearity of the system as a whole.

Table 3.5: Convergence Data

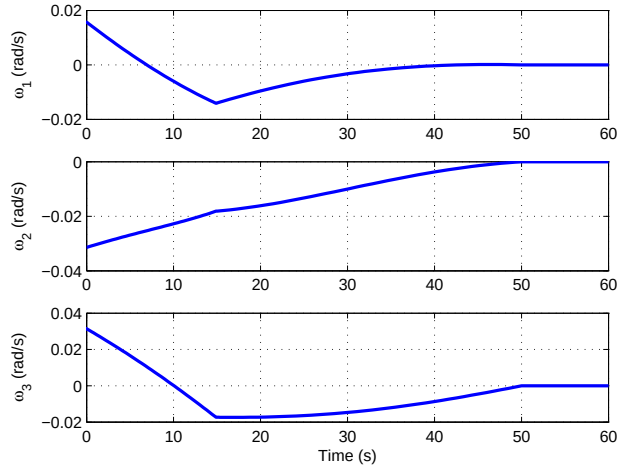
Iter.	$\mathbf{N}_{w0}^T$ (Nm)	$\mathbf{N}_{wT}^T$ (Nm)	t_T (s)	$\mathbf{q}_v$ Error
0	$\begin{bmatrix} 0.1000 & 0.1000 & 0.1000 \end{bmatrix}$	$\begin{bmatrix} -0.1000 & -0.1000 & -0.1000 \end{bmatrix}$	15.0	$\begin{bmatrix} 0.6304 & -0.3972 & 0.4652 \end{bmatrix}$
8	$\begin{bmatrix} 0.4234 & -0.1335 & 0.4386 \end{bmatrix}$	$\begin{bmatrix} -0.0001 & -0.0331 & -0.1422 \end{bmatrix}$	14.8750	$\begin{bmatrix} -0.0000 & -0.0000 & 0.0000 \end{bmatrix}$

3.1.2.3 Numerical Example 3: Stabilization from Initial Rotation

The goal of the targeter in this example is to stabilize the spacecraft from a higher initial rotation rate; the final attitude is unconstrained. As discussed in the previous example, simultaneously targeting both a desired final attitude and terminal angular velocity is difficult if the spacecraft is already rotating at a rate of about 3° per second or greater. However, this issue can



(a) Attitude



(b) Angular Velocity

Figure 3.6: Attitude and Angular Velocity, Example 2

be overcome by applying the targeting algorithm sequentially. First, as shown in this example, only the final angular velocity is targeted in order to drive the rotation rate to zero. This gives a final converged state with zero angular velocity and an arbitrary attitude. That final converged state is then used as the initial state for targeting the desired attitude and angular velocity, as demonstrated previously. In the following example, the vehicle has an initial rotation rate of approximately 30° per second that the targeter must nullify by $t_f = 50$ s. The applied wheel torque at t_T is now assumed to be $\mathbf{0}$, so the control parameters are $\mathbf{N}_{w0}$ and t_T. As the results in Table 3.6 and Figure 3.7 illustrate, the algorithm converges in 10 iterations to the appropriate control values, zeroing out the initial rotation. The required initial torque, unsurprisingly, is significantly greater than in the previous examples; because the initial angular momentum of the system is higher, the rate of momentum transfer to the wheels much also be much higher in order to meet the target angular velocity within the specified time period.

Table 3.6: Convergence Data

Iter.	$\mathbf{N}_{w0}^T$ (Nm)	t_T (s)	$\boldsymbol{\omega}$ Error (rad/s)
0	$\begin{bmatrix} 0.5 & 0.5 & 0.5 \end{bmatrix}$	30.0	$\begin{bmatrix} -0.1319 & 0.4474 & -0.2236 \end{bmatrix}$
10	$\begin{bmatrix} 0.9178 & -1.9665 & 1.3793 \end{bmatrix}$	31.1169	$\begin{bmatrix} -0.0000 & 0.0000 & -0.0000 \end{bmatrix}$

It should be noted that the final augmented state in this example is not entirely analogous to the initial state in the first example, despite the zero angular velocity of the vehicle. In this case, the wheel angular momentum $\mathbf{h}_w$ has a nonzero starting value, and so the cross terms in Equation 3.14 will

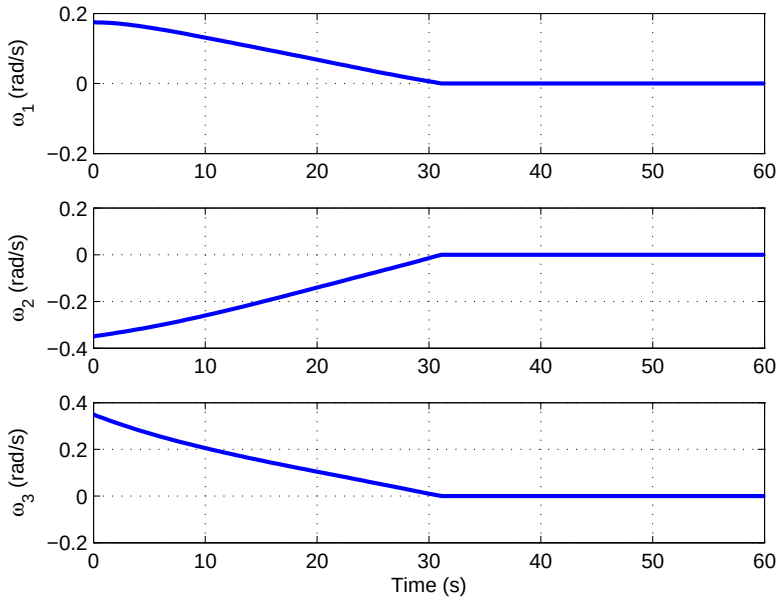


Figure 3.7: Angular Velocity, Example 3

still have a significant effect on the system dynamics. This in turn increases the nonlinearity of the system, causing difficulty for the linear targeter just as the nonzero initial angular velocity does. However, if it is assumed that the vehicle employs an ancillary control system for momentum dumping[17], then the value of $\mathbf{h}_w$ can be negated prior to initiating the attitude targeting. With both $\boldsymbol{\omega}$ and $\mathbf{h}_w$ starting at $\mathbf{0}$, the problem becomes equivalent to the first example in this section.

3.1.3 Level I Targeting Example: Finite Burn

Consider a trajectory segment in the two-level targeter framework defined by patch points $k-1$ and k, as shown in Fig. 3.8. With the addition of finite burn subarcs, the equations of motion become

$$\dot{\mathbf{x}}(t) = \tilde{\mathbf{f}}(\mathbf{x}(t), t, m(t), \dot{m}_g(t), \tilde{\mathbf{u}}(t)), \tag{3.29}$$

where $m(t)$ is the mass of the vehicle, $\dot{m}_g(t)$ is the propellant flow rate, and $\tilde{\mathbf{u}}(t)$ is the full thrust vector (direction and magnitude). In the context of Equation 3.1, these terms ($m(t)$, $\dot{m}_g(t)$, and $\tilde{\mathbf{u}}(t)$) are the elements of $\boldsymbol{\zeta}(t)$. For this investigation, $\tilde{\mathbf{u}}(t)$ is assumed to be either constant or linearly varying for $t_j \leq t < t_{T_j}$ and zero for $t_{T_j} < t \leq t_{j+1}$, where T, as before, denotes the point at which the actuation is turned off (i.e. engine cutoff, in this example). Note that the j subscripts indicate the point at which the j^{th} finite burn maneuver occurs; they do not necessarily represent sequential patch points. The equation

for $\tilde{\mathbf{u}}(t)$ over the entire trajectory is

$$\tilde{\mathbf{u}}(t) = \sum_{j=1}^{n_{\Delta \mathbf{v}}} \tilde{\mathbf{u}}_j \left[1(t - t_j) - 1(t - t_{T_j}) \right], \tag{3.30}$$

where $\tilde{\mathbf{u}}_j$ is the thrust vector at maneuver point j, $1(t)$ is the unit step function, and $n_{\Delta \mathbf{v}}$ represents the total number of finite burn maneuvers along the trajectory. A plot of the thrust magnitude for a 3-burn sequence is shown in Fig. 3.9.

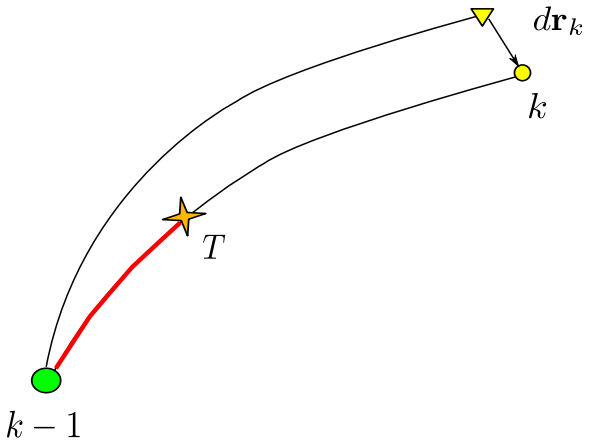


Figure 3.8: Level I Process

For the finite burn case, the augmented state vector is

$$\mathbf{x}_{k-1}^{+} = \begin{bmatrix} \mathbf{r}_{k-1} \\ \mathbf{v}_{k-1}^{+} \\ m_{k-1} \\ \dot{m}_{g_{k-1}} \\ \mathbf{u}_{k-1} \end{bmatrix}, \tag{3.31}$$

where m_{k-1} and $\dot{m}_{g_{k-1}}$ represent the spacecraft mass and the propellant flow rate associated with patch point $k-1$, respectively. The variable $\mathbf{u}_{k-1}$ can be defined as either the full thrust vector $\tilde{\mathbf{u}}_{k-1}$ or as a vector describing only the

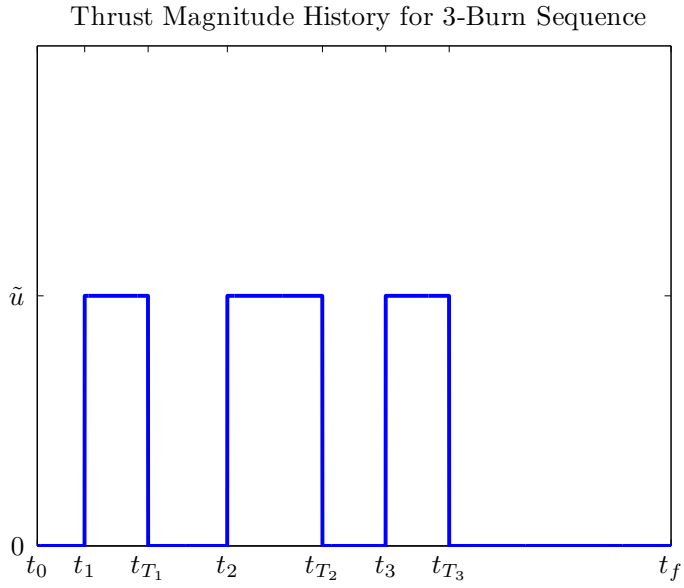


Figure 3.9: Burn Sequence

direction of the thrust[44]. In this work, $\mathbf{u}_{k-1}$ is assumed to represent only the thrust direction. The algorithm targets the terminal position vector at point k ($\mathbf{r}_k$), and the control variables are the vector $\mathbf{u}_{k-1}$, representing either the full thrust or thrust direction, and the cutoff time of the burn (t_T). Under this formulation there are 4 control parameters with which to target the 3 elements of $\mathbf{r}_k$. The variational equation for the burn subsegment is

$$\begin{bmatrix} d\mathbf{r}_T - \mathbf{v}_T^- dt_T \\ d\mathbf{v}_T^- - \mathbf{a}_T^- dt_T \\ dm_T^- + \dot{m}_{g_T}^- dt_T \\ d\dot{m}_{g_T}^- - \ddot{m}_{g_T}^- dt_T \\ d\mathbf{u}_T^- - \dot{\mathbf{u}}_T^- dt_T \end{bmatrix} = \Phi(T, k-1) \begin{bmatrix} d\mathbf{r}_{k-1} - \mathbf{v}_{k-1}^+ dt_{k-1} \\ d\mathbf{v}_{k-1}^+ - \mathbf{a}_{k-1}^+ dt_{k-1} \\ dm_{k-1}^+ + \dot{m}_{g_{k-1}}^+ dt_{k-1} \\ d\dot{m}_{g_{k-1}}^+ - \ddot{m}_{g_{k-1}}^+ dt_{k-1} \\ d\mathbf{u}_{k-1}^+ - \dot{\mathbf{u}}_{k-1}^+ dt_{k-1} \end{bmatrix}, \qquad (3.32)$$

where $\Phi(T, k-1)$ is the state transition matrix between patch point k and point

T. As in the previous formulations, the state transition matrix is partitioned into sub-matrices corresponding to each state:

$$\Phi(T,k-1) = \begin{bmatrix} A_{T,k-1} & B_{T,k-1} & E_{T,k-1} & F_{T,k-1} & G_{T,k-1} \\ C_{T,k-1} & D_{T,k-1} & H_{T,k-1} & I_{T,k-1} & J_{T,k-1} \\ K_{T,k-1} & L_{T,k-1} & M_{T,k-1} & N_{T,k-1} & O_{T,k-1} \\ P_{T,k-1} & Q_{T,k-1} & R_{T,k-1} & S_{T,k-1} & T_{T,k-1} \\ U_{T,k-1} & V_{T,k-1} & W_{T,k-1} & X_{T,k-1} & Y_{T,k-1} \end{bmatrix}. \tag{3.33}$$

For the subsequent coasting subsegment, the variational equation, with partitioned state transition matrix, takes the same form as in the impulsive formulation,[25]

$$\begin{bmatrix} d\mathbf{r}_k - \mathbf{v}_k^- dt_k \\ d\mathbf{v}_k^- - \mathbf{a}_k^- dt_k \end{bmatrix} = \begin{bmatrix} A_{k,T} & B_{k,T} \\ C_{k,T} & D_{k,T} \end{bmatrix} \begin{bmatrix} d\mathbf{r}_T - \mathbf{v}_T^+ dt_T \\ d\mathbf{v}_T^+ - \mathbf{a}_T^+ dt_T \end{bmatrix} \tag{3.34}$$

Here, both the initial and final times of the arc (t_{k-1} and t_k, respectively) are fixed, though that is not a requirement. The initial position $\mathbf{r}_{k-1}$, velocity $\mathbf{v}_{k-1}^+$, and mass m_{k-1}^+ are also fixed, as well as the mass flow rate, $\dot{m}_{g_{k-1}}^+$. It is important to note that $\mathbf{v}_T^+ = \mathbf{v}_T^-$ (and therefore $d\mathbf{v}_T^+ = d\mathbf{v}_T^-$). Furthermore, $d\mathbf{v}_T^+ - \mathbf{a}_T^+ dt_T = d\mathbf{v}_T^- - \mathbf{a}_T^- dt_T + (\mathbf{a}_T^- - \mathbf{a}_T^+) dt_T$. Incorporating these substitutions, the first two vector variational equations from Equation 3.32 and Equation 3.34 can be combined to give an expression for $d\mathbf{r}_k$:

$$d\mathbf{r}_k = \begin{bmatrix} (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) & B_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+) \end{bmatrix} \begin{bmatrix} d\mathbf{u}_{k-1}^+ \\ dt_T \end{bmatrix} \tag{3.35}$$

As in the impulsive Level I method,[25] a minimum norm solution is selected to obtain the desired change in the control variables,

$$\begin{bmatrix} d\mathbf{u}_{k-1}^+ \\ dt_T \end{bmatrix} = \tilde{M}^T (\tilde{M}\tilde{M}^T)^{-1} d\mathbf{r}_k, \tag{3.36}$$

where $\tilde{M}$, the Level I State Relationship Matrix (SRM) is given by

$$\tilde{M} = \left[\begin{array}{cc} (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) & B_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+) \end{array} \right]. \tag{3.37}$$

A minimum norm solution identifies the smallest change in the control parameters, in this case $d\mathbf{u}_{k-1}^+$ and dt_T, that lead to the desired changes in the constraint errors. Of course, these corrections are linear in nature and, as such, an iterative process is required to converge on the specified constraints in the nonlinear system.

A potential issue that can occur with this formulation is that the converged burn cutoff time t_T may fall after the terminal segment time t_k. To resolve this, t_k becomes a control parameter (and is thus allowed to vary), and an additional constraint is appended to enforce $t_T \leq t_k$. This constraint, $\alpha_t = t_T - t_k$, would only be active if t_T is greater than t_k. The Level I constraint equation then becomes

$$\left[\begin{array}{c} d\mathbf{r}_k \\ d\alpha_t \end{array} \right] = \left[\begin{array}{ccc} \frac{\partial \mathbf{r}_k}{\partial \mathbf{u}_{k-1}^+} & \frac{\partial \mathbf{r}_k}{\partial t_T} & \frac{\partial \mathbf{r}_k}{\partial t_k} \\ \frac{\partial \alpha_t}{\partial \mathbf{u}_{k-1}^+} & \frac{\partial \alpha_t}{\partial t_T} & \frac{\partial \alpha_t}{\partial t_k} \end{array} \right] \left[\begin{array}{c} d\mathbf{u}_{k-1}^+ \\ dt_T \\ dt_k \end{array} \right], \tag{3.38}$$

and the three control parameters are again found using the minimum norm solution. In practice, this situation can be avoided through proper patch point selection; the duration of a segment containing a burn arc should always be significantly greater than the expected burn time.

To determine an initial guess for the finite burn parameters (i.e. thrust direction, $\mathbf{u}_{k-1}$, and burn cutoff time, t_T), the impulsive Level I process is first used to compute an impulsive correction. The impulsive $\Delta\mathbf{v}$ direction is used

as an initial guess for the thrust direction. If $\mathbf{u}_{k-1}$ is defined as the full thrust vector, then the initial guess for the thrust magnitude is the given thrust of the engine. Finally, an initial guess for the burn duration is deduced using the rocket equation,

$$\Delta \mathrm{v}_{k-1} = -I_{sp} g_0 ln \left(1 - \frac{\dot{m}_{g_{k-1}} \Delta t_{burn}}{m_{k-1}}\right), \tag{3.39}$$

and rearranging to obtain

$$\Delta t_{burn} = \frac{m_{k-1}}{\dot{m}_{g_{k-1}}} \left(1 - e^{\frac{-\Delta \mathrm{v}_{k-1} \dot{m}_{g_{k-1}}}{u_{k-1}}}\right) = \frac{m_{k-1}}{\dot{m}_{g_{k-1}}} \left(1 - e^{\frac{-\Delta \mathrm{v}_{k-1}}{I_{sp} g_0}}\right) \tag{3.40}$$

From the burn duration, the cutoff time is calculated as $dt_T = dt_{k-1} + \Delta t_{burn}$. This is similar to the Lambert guidance approach used in missile targeting[63].

The impulsive burn approximation becomes less valid as the finite burn time increases. Therefore, the terminal error after the first iteration can be very large when the burn duration is long. If the burn direction is assumed to be constant throughout the entire maneuver, small errors in direction can be greatly magnified by the end of a long burn. Thus, it is advantageous to incorporate an optional steering law into the finite burn algorithm.

3.1.3.1 Level I Finite Burn with Linear Steering

A linear steering law can be easily implemented into the finite burn application of the targeting algorithm. If a constant $\dot{\mathbf{u}}_{k-1}$ is allowed in the thrust equation, such that

$$\mathbf{u}(t) = \mathbf{u}_{k-1} + \dot{\mathbf{u}}_{k-1}(t - t_{k-1}), \tag{3.41}$$

the augmented state vector becomes $\mathbf{x}_{k-1} = \left[\mathbf{r}_{k-1} \ \mathbf{v}_{k-1} \ m_{k-1} \ \dot{m}_{g_{k-1}} \ \mathbf{u}_{k-1} \ \dot{\mathbf{u}}_{k-1} \right]^T$. The Level I control variables are now $d\mathbf{u}_k, d\dot{\mathbf{u}}_k$ and thrust cutoff time dt_T. The derivation for both the Level I and Level II processes follows the same procedure as the constant steering algorithm, with additional terms resulting from the nonzero $\dot{\mathbf{u}}_{k-1}$. With a linear steering law, the partitioned STM for the burn subarc is

$$\Phi(T,k-1) = \begin{bmatrix} A_{T,k-1} & B_{T,k-1} & E_{T,k-1} & F_{T,k-1} & G_{T,k-1} & \Gamma_{T,k-1} \\ C_{T,k-1} & D_{T,k-1} & H_{T,k-1} & I_{T,k-1} & J_{T,k-1} & \Lambda_{T,k-1} \\ K_{T,k-1} & L_{T,k-1} & M_{T,k-1} & N_{T,k-1} & O_{T,k-1} & \Xi_{T,k-1} \\ P_{T,k-1} & Q_{T,k-1} & R_{T,k-1} & S_{T,k-1} & T_{T,k-1} & \Psi_{T,k-1} \\ U_{T,k-1} & V_{T,k-1} & W_{T,k-1} & X_{T,k-1} & Y_{T,k-1} & \Omega_{T,k-1} \\ U_{LS_{T,k-1}} & V_{LS_{T,k-1}} & W_{LS_{T,k-1}} & X_{LS_{T,k-1}} & Y_{LS_{T,k-1}} & \Omega_{LS_{T,k-1}} \end{bmatrix}. \tag{3.42}$$

The subsequent change to the control variables is given by

$$\begin{bmatrix} d\mathbf{u}_{k-1}^+ \\ d\dot{\mathbf{u}}_{k-1}^+ \\ dt_T \end{bmatrix} = \tilde{M}^T (\tilde{M}\tilde{M}^T)^{-1} d\mathbf{r}_k, \tag{3.43}$$

and the SRM $\tilde{M}$ is

$$\tilde{M} = \begin{bmatrix} (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) & (A_{k,T}\Gamma_{T,k-1} + B_{k,T}\Lambda_{T,k-1}) & B_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+) \end{bmatrix}. \tag{3.44}$$

When implemented numerically, it is possible for the steering rate to increase too quickly, causing degradation in the algorithm performance. To avoid excessive growth of $\dot{\mathbf{u}}$, variable scaling is often useful when enabling linear steering.

A similar approach may be used to implement more complex steering laws, effectively allowing the targeter to incorporate the functions of a guidance law into the solution process. The determination of the thrust direction

throughout the duration of the burn is typically accomplished as a separate task, using the $\Delta\mathbf{v}$ vector provided by the targeting algorithm[40, ?, ?]. With this continuous actuation targeting algorithm, however, the guidance problem is solved concurrently with the targeting.

3.1.3.2 Level I Earth Entry Targeting

The following example shows the finite burn Level I algorithm implemented for a trajectory segment from near the Moon to a specified Earth entry interface (EEI) position. In keeping with the typical structure of the two-level corrector, only the terminal position vector is targeted for this example. Tables 3.7 and 3.8 list the initial conditions and target position for this problem. Two cases are examined; in the first, the available engine has sufficiently high thrust that the burn duration is small and the impulsive approximation of the maneuver can be assumed to be relatively accurate. For the second case, a much lower-thrust engine is used, so that the impulsive maneuver model breaks down. The engines are modeled after the main and auxiliary engines onboard the Orion Multi-Purpose Crew Vehicle (MPCV)[44], and the engine parameters for each case are provided in Table 3.9.

For a burn of finite duration, the desired burn direction may not be the same at the end of the burn as it is at the beginning of the burn, particularly as the integration time increases. If the direction of the burn is assumed to be inertially fixed, and is initially aligned with the $\Delta\mathbf{v}$ vector computed by the impulsive targeter, then the error in the thrust direction by the end

Table 3.7: Initial Conditions

	Initial Value
Epoch	6-Apr-2024 10:07:23.8635 TDT
Mass (kg)	20339.9 (total fuel = 8063.65 kg)
x (km)	347852.0598074741
y (km)	-81443.4055313879
z (km)	-54980.6777783854
$\dot{x}$ (km/s)	0.146041240182201
$\dot{y}$ (km/s)	0.609351260611497
$\dot{z}$ (km/s)	-0.888636728804498

Table 3.8: Target Terminal State

Constraint	Value
x (km)	-5886.2551
y (km)	2742.2150
z (km)	217.3391
Time of Flight (s)	342985.2619989844

Table 3.9: Engine Parameters

Engine	Thrust (N)	I_{sp} (s)
Main (High Thrust)	33,361.6621	326
Auxiliary (Low Thrust)	4,448.0	309

Table 3.10: Level I EEI Targeting Results

	High Thrust	Low Thrust
Initial Position Error (km)	2.14187e5	2.03681e5
Final Position Error (km)	3.58942e-5	1.39025e-4
Burn Duration (s)	232.5476	2367.8813
Total Δv (km/s)	0.4062	0.5679
Iterations	64	85

of a long burn can be substantial. To combat this, when the lower-thrust engine is employed, the linear steering law described in the previous section is implemented as well. Using fixed-direction thrust, the targeter is unable to converge for the low-thrust engine.

Table 3.10 lists the convergence data for both scenarios examined. With the high-thrust engine, the burn duration is short and the Δv for the maneuver is, as expected, very similar to the impulsive Δv value of 0.4047 km/s. By contrast, the low-thrust engine maneuver has a significantly higher Δv due to its extremely long burn duration - over 10 times that of the high-thrust engine. Interestingly, however, the initial error at the target position is actually slightly smaller for the low-thrust engine with linear steering employed. This suggests that, even for the high-thrust case, the thrust direction error at the termination of the burn is causing inaccuracies in the final position.

In general, both cases considered here required a substantially increased number of iterations relative to later examples. This is attributed to the sensitivities traditionally associated with a Level I process. Specifically, the success of a Level I process is sensitive to the integration time. Since this

particular example employs a Level I process to transfer the vehicle from the vicinity of the moon to the entry interface at Earth, the time of flight is too long for the number of control parameters available. Here, a Level II process[25] becomes useful because it allows for an increased number of control parameters and also the ability to incorporate an arbitrary number of constraints. The development of a Level II process that accommodates continuous actuation is discussed next.

3.2 Level II Process

The continuous control Level II process, like the impulsive Level II correction, uses the positions and times of the patch points as control variables. In the classical two-level corrector,[25] velocity discontinuities between coast segments arise due to the Level I process. This is also applicable to the continuous formulation, except at the point where a control input is initiated. Here, the controlled segment is always assumed to start with the same initial velocity as the terminal velocity of the preceding arc. Thus, a velocity discontinuity can occur, during the Level I process, at the point where the coast subarc, as defined in Fig. 3.8, joins with the following trajectory segment. Although this problem at first seems identical to the impulsive maneuver targeting, since the velocity discontinuity falls between two coast arcs, the partial derivatives for the incoming terminal velocity of the segment with respect to the control variables differ due to the actuation segment at the beginning of the arc. In the most general form, the partials for any Level II constraint $\boldsymbol{\alpha}$ (including

discontinuities) will have an additional term representing the indirect relationship between the constraint and any Level II control β through the actuation parameters $\boldsymbol{\zeta}$ and t_T, such that

$$\frac{d\boldsymbol{\alpha}}{d\beta} = \frac{\partial \boldsymbol{\alpha}}{\partial \beta} + \frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{\zeta}}\frac{\partial \boldsymbol{\zeta}}{\partial \beta} + \frac{\partial \boldsymbol{\alpha}}{\partial t_T}\frac{\partial t_T}{\partial \beta}. \tag{3.45}$$

To illustrate how this may be specifically implemented for a given system, the remainder of this section will focus on the derivation of the Level II process in the context of an orbital transfer problem with finite burn maneuvers.

Recall from the Level I formulation in Section 3.1.3 that $\mathbf{v}_T^- = \mathbf{v}_T^+$ at the terminal point of the burn arc and thus that $d\mathbf{v}_T^+ - \mathbf{a}_T^+ dt_T = d\mathbf{v}_T^- - \mathbf{a}_T^- dt_T + (\mathbf{a}_T^- - \mathbf{a}_T^+)dt_T$. For the Level II process, $dm_{k-1}^+ = 0$, $\ddot{m}_{g_{k-1}}^+ = 0$, and $\dot{\mathbf{u}}_{k-1}^+ = \mathbf{0}$. It is still assumed that $\dot{m}_g$ is a fixed constant, i.e. $d\dot{m}_{gk-1}^+ = 0$. Note also that the terminal time of the burn at patch point $k-1$, $t_{T_{k-1}}$, is directly related to the time at patch point $k-1$: $t_{T_{k-1}} = t_{k-1} + \Delta t_{burn_{k-1}}$, where Δt_{burn} is the duration of the burn and is held constant in the Level II process. Using these relationships and assumptions, along with Equations 3.32 and 3.34, an expression is found for $d\mathbf{v}_k^-$ in terms of the state at patch point $k-1$ and the state transition matrix,

$$\begin{aligned} d\mathbf{v}_k^- = {} & C_{k,T}[A_{T,k-1}(d\mathbf{r}_{k-1} - \mathbf{v}_{k-1}^+ dt_{k-1}) \\ & + B_{T,k-1}(d\mathbf{v}_{k-1}^+ - \mathbf{a}_{k-1}^+ dt_{k-1}) + E_{T,k-1}\dot{m}_{g_{k-1}}^+ dt_{k-1} + G_{T,k-1}d\mathbf{u}_{k-1}^+] \\ & + D_{k,T}[C_{T,k-1}(d\mathbf{r}_{k-1} - \mathbf{v}_{k-1}^+ dt_{k-1}) + D_{T,k-1}(d\mathbf{v}_{k-1}^+ - \mathbf{a}_{k-1}^+ dt_{k-1}) \\ & + H_{T,k-1}\dot{m}_{g_{k-1}}^+ dt_{k-1} + J_{T,k-1}d\mathbf{u}_{k-1}^+ + (\mathbf{a}_T^- - \mathbf{a}_T^+)(dt_{k-1} + \Delta t_{burn})] + \mathbf{a}_k^- dt_k. \end{aligned} \tag{3.46}$$

In order to write $d\mathbf{v}_k^-$ only in terms of the Level II control variables, the first vector equation from Equation 3.32 is used to solve for $d\mathbf{v}_{k-1}^+$, $d\mathbf{u}_{k-1}^+$, and Δt_{burn} in terms of the Level II controls. From the minimum norm solution,

$$\begin{aligned}\begin{bmatrix} d\mathbf{v}_{k-1}^+ \\ d\mathbf{u}_{k-1}^+ \\ \Delta t_{burn} \end{bmatrix} &= Z^T(ZZ^T)^{-1}[d\mathbf{r}_k - \mathbf{v}_k^- dt_k \\ &- (A_{k,T}A_{T,k-1} + B_{k,T}C_{T,k-1})(d\mathbf{r}_{k-1} - \mathbf{v}_{k-1}^+ dt_{k-1}) \\ &+ (A_{k,T}B_{T,k-1} + B_{k,T}D_{T,k-1})\mathbf{a}_{k-1}^+ dt_{k-1} \\ &- (A_{k,T}E_{T,k-1} + B_{k,T}H_{T,k-1})\dot{m}_{g_{k-1}}^+ dt_{k-1} \\ &- B_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+)dt_{k-1}].\end{aligned} \tag{3.47}$$

where

$$Z = \left[(A_{k,T}B_{T,k-1} + B_{k,T}D_{T,k-1})\ (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1})\ B_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+) \right]. \tag{3.48}$$

With this expression, the partial derivatives of $\Delta\mathbf{v}_k$ with respect to each control variable can be found using the same method as in the impulsive formulation. Let

$$\tilde{Z} = \left[(C_{k,T}B_{T,k-1} + D_{k,T}D_{T,k-1})\ (C_{k,T}G_{T,k-1} + D_{k,T}J_{T,k-1})\ D_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+) \right] Z^T(ZZ^T)^{-1}. \tag{3.49}$$

Then, because it is assumed that the arc from patch point k to $k+1$ is a coast

arc, the partial derivatives of $\Delta\mathbf{v}_k$ are

$$\begin{aligned}
\frac{\partial \Delta\mathbf{v}_k}{\partial \mathbf{r}_{k-1}} &= -[(C_{k,T}A_{T,k-1} + D_{k,T}C_{T,k-1}) - \tilde{Z}(A_{k,T}A_{T,k-1} + B_{k,T}C_{T,k-1})], \\
\frac{\partial \Delta\mathbf{v}_k}{\partial t_{k-1}} &= -([(C_{k,T}E_{T,k-1} + D_{k,T}H_{T,k-1}) - \tilde{Z}(A_{k,T}E_{T,k-1} + B_{k,T}H_{T,k-1})]\dot{m}^+_{g_{k-1}} \\
&\quad - [(C_{k,T}B_{T,k-1} + D_{k,T}D_{T,k-1}) - \tilde{Z}(A_{k,T}B_{T,k-1} + B_{k,T}D_{T,k-1})]\mathbf{a}^+_{k-1} \\
&\quad - [(C_{k,T}A_{T,k-1} + D_{k,T}C_{T,k-1}) - \tilde{Z}(A_{k,T}A_{T,k-1} + B_{k,T}C_{T,k-1})]\mathbf{v}^+_{k-1} \\
&\quad + (D_{k,T} - \tilde{Z}B_{k,T})(\mathbf{a}^-_T - \mathbf{a}^+_T)), \\
\frac{\partial \Delta\mathbf{v}_k}{\partial \mathbf{r}_k} &= -B^{-1}_{k+1,k}A_{k+1,k} - \tilde{Z}, \\
\frac{\partial \Delta\mathbf{v}_k}{\partial t_k} &= B^{-1}_{k+1,k}A_{k+1,k}\mathbf{v}^+_k + \mathbf{a}^+_k - (\mathbf{a}^-_k - \tilde{Z}\mathbf{v}^-_k), \\
\frac{\partial \Delta\mathbf{v}_k}{\partial \mathbf{r}_{k+1}} &= B^{-1}_{k+1,k}, \\
\frac{\partial \Delta\mathbf{v}_k}{\partial t_{k+1}} &= -B^{-1}_{k+1,k}\mathbf{v}^-_{k+1}.
\end{aligned} \tag{3.50}$$

Since the position and time at $k+1$ do not affect the thrust parameters at $k-1$, the partial derivatives for these controls are the same as in the impulsive formulation. The above partials are employed in the standard level II process[25], replacing the corresponding impulsive partials in the Level II SRM. To incorporate linear steering into the Level II process, it is only necessary to include additional terms in the Z, $\tilde{Z}$, and $\frac{\partial \Delta\mathbf{v}_k}{\partial t_{k-1}}$ expressions:

$$Z = \begin{bmatrix} (A_{k,T}B_{T,k-1} + B_{k,T}D_{T,k-1}) \\ (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) \\ (A_{k,T}\Gamma_{T,k-1} + B_{k,T}\Lambda_{T,k-1}) \\ B_{k,T}(\mathbf{a}^-_T - \mathbf{a}^+_T) \end{bmatrix}^T . \tag{3.51}$$

$$\tilde{Z} = \begin{bmatrix} (C_{k,T}B_{T,k-1} + D_{k,T}D_{T,k-1}) \\ (C_{k,T}G_{T,k-1} + D_{k,T}J_{T,k-1}) \\ (C_{k,T}\Gamma_{T,k-1} + D_{k,T}\Lambda_{T,k-1}) \\ D_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+) \end{bmatrix}^T Z^T(ZZ^T)^{-1}. \tag{3.52}$$

$$\frac{\partial \Delta \mathbf{v}_k}{\partial t_{k-1}} = \frac{\partial \Delta \mathbf{v}_k}{\partial t_{k-1}}_{const} + \left[(C_{k,T}G_{T,k-1} + D_{k,T}J_{T,k-1}) - \tilde{Z}(A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1})\right] \dot{\mathbf{u}}_{k-1}. \tag{3.53}$$

The rest of the Level II correction proceeds exactly as before.

3.2.1 Maneuver Sum Constraint

In addition to the velocity continuity constraint, endpoint and interior path constraints may be imposed during the Level II process[59]. One such constraint in the finite thrust application is on the total Δv sum of the maneuvers. The finite burn formulation of this constraint is based on the impulsive maneuver sum constraint[46], which takes the form

$$\alpha_{\Delta \mathrm{v}} = \sum_{j=1}^{n_{\Delta \mathbf{v}}} \Delta \mathrm{v}_j \leq \Delta \mathrm{v}_{\max}. \tag{3.54}$$

Only the composition of the associated partial derivatives and the error calculation changes.

To derive the burn maneuver constraint, it is necessary to determine the partial derivatives of the magnitude of $\Delta \mathbf{v}_k$, i.e. the maneuver that results from the burn at patch point k, with respect to the Level II control variables. The rocket equation (Equation 3.39) is again used to determine $\Delta \mathrm{v}_k$ in terms of the state parameters. Because I_{sp}, g_0, and $\dot{m}_{g_k}$ are fixed, the cost of a

maneuver at patch point k depends on the duration of the burn at k and the initial mass. The burn duration Δt_{burn} and mass m_k are in turn functions of the control variables, and the chain rule is used to determine the partials of $\Delta \mathrm{v}_k$ with respect to the controls. The partial derivative of $\Delta \mathrm{v}_k$ with respect to Δt_{burn} at patch point k is given by

$$\frac{\partial \Delta \mathrm{v}_k}{\partial \Delta t_{burn}} = I_{sp} g_0 \left(\frac{m_k}{m_k - \dot{m}_{g_k} \Delta t_{burn}} \right) \left(\frac{\dot{m}_{g_k}}{m_k} \right). \tag{3.55}$$

With respect to m_k, the partial derivative is

$$\frac{\partial \Delta \mathrm{v}_k}{\partial m_k} = \frac{I_{sp} g_0}{m_k} \left(\frac{\dot{m}_{g_k} \Delta t_{burn}}{m_k - \dot{m}_{g_k} \Delta t_{burn}} \right). \tag{3.56}$$

The partial derivatives of Δt_{burn} with respect to the control variables are determined using the variational equations from points $k-1$ to k, k to T (the termination of the burn segment), and $k+1$ to T. Recalling that $\Delta t_{burn} = t_T - t_k$, the partials are found to be

$$\begin{aligned}
\frac{\partial \Delta t_{burn}}{\partial \mathbf{r}_{k-1}} &= -\frac{\hat{u}_k^T}{||\Delta \mathbf{a}_T||} (D_{T,k} B_{k-1,k}^{-1} - \tilde{S} B_{T,k} B_{k-1,k}^{-1}), \\
\frac{\partial \Delta t_{burn}}{\partial t_{k-1}} &= \frac{\hat{u}_k^T}{||\Delta \mathbf{a}_T||} (D_{T,k} B_{k-1,k}^{-1} - \tilde{S} B_{T,k} B_{k-1,k}^{-1}) \mathbf{v}_{k-1}^+, \\
\frac{\partial \Delta t_{burn}}{\partial \mathbf{r}_k} &= -\frac{\hat{u}_k^T}{||\Delta \mathbf{a}_T||} [(C_{T,k} + D_{T,k} D_{k,k-1} B_{k,k-1}^{-1}) - \tilde{S} (A_{T,k} + B_{T,k} D_{k,k-1} B_{k,k-1}^{-1})], \\
\frac{\partial \Delta t_{burn}}{\partial t_k} &= \frac{\hat{u}_k^T}{||\Delta \mathbf{a}_T||} ([(C_{T,k} + D_{T,k} D_{k,k-1} B_{k,k-1}^{-1}) - \tilde{S} (A_{T,k} + B_{T,k} D_{k,k-1} B_{k,k-1}^{-1})] \mathbf{v}_k^- \\
&\quad - (D_{T,k} - \tilde{S} B_{T,k})(\mathbf{a}_k^- - \mathbf{a}_k^+) - (H_{T,k} - \tilde{S} E_{T,k}) \dot{m}_g), \\
\frac{\partial \Delta t_{burn}}{\partial \mathbf{r}_{k+1}} &= \frac{\hat{u}_k^T}{||\Delta \mathbf{a}_T||} (C_{T,k+1} - \tilde{S} A_{T,k+1}), \\
\frac{\partial \Delta t_{burn}}{\partial t_{k+1}} &= -\frac{\hat{u}_k^T}{||\Delta \mathbf{a}_T||} [(C_{T,k+1} - \tilde{S} A_{T,k+1}) \mathbf{v}_{k+1}^- + (D_{T,k+1} - \tilde{S} B_{T,k+1}) \mathbf{a}_{k+1}^-],
\end{aligned} \tag{3.57}$$

where $S = \begin{bmatrix} -G_{T,k} & B_{T,k+1} \end{bmatrix}$ and $\tilde{S} = \begin{bmatrix} -J_{T,k} & D_{T,k+1} \end{bmatrix} S^T(SS^T)^{-1}$. It is more complicated to determine the partials of m_k with respect to the control variables. Since $\dot{m}_g$ is a fixed, constant value, m_k depends only on the previous burn durations. Thus, the initial mass at the beginning of a maneuver will have a dependence on the positions and times associated with any previous maneuvers that have occurred, as shown above. Using the chain rule, the final form of the partial derivative of the constraint α (the sum of all the burn Δvs) with respect to any control variable β_k in the set of control variables associated with patch point k is

$$\frac{\partial \alpha}{\partial \beta_k} = \sum_{n=1}^{N_{\Delta \mathrm{v}}} \frac{\partial \Delta \mathrm{v}_n}{\partial \beta_k} \tag{3.58}$$

where $N_{\Delta \mathrm{v}}$ is the total number of maneuvers implemented along the trajectory and

$$\frac{\partial \Delta \mathrm{v}_n}{\partial \beta_k} = \frac{\partial \Delta \mathrm{v}_n}{\partial \Delta t_{burn_n}} \frac{\partial \Delta t_{burn_n}}{\partial \beta_k} + \frac{\partial \Delta \mathrm{v}_n}{\partial m_n} \frac{\partial m_n}{\partial \beta_k}. \tag{3.59}$$

Because the mass at the time of a burn, m_n, depends on the propellant mass consumed during the previous burns,

$$\frac{\partial m_n}{\partial \beta_k} = \frac{\partial m_{n-1}}{\partial \beta_k} - \dot{m}_{g_{n-1}} \frac{\partial \Delta t_{burn_{n-1}}}{\partial \beta_k}. \tag{3.60}$$

A similar relationship exists for the remaining mass partials (m_{n-2} to m_1) with respect to β_1. These partials are then employed during the Level II process.[25]

3.3 Sample Finite Burn Applications: Trans-Earth Injection (TEI) Simulation and Results

Over the course of the following section, the finite burn targeting algorithm is tested and validated through a precision entry application, which involves a lunar return trajectory and the associated three-burn trans-earth injection (TEI) sequence. The goal is to identify the maneuver sequence that meets the specified set of lunar departure and Earth entry interface constraints without violating the available fuel budget. The performance of the finite burn algorithm is contrasted against that of the impulsive algorithm[44].

For each case considered, the parameters that define the initial low lunar orbit departure are the same. These conditions are listed in Table 3.11; engine parameters for the finite burn maneuvers are the same as in Table 3.9. Components of the initial state in Table 3.11 are in the J2000 Moon-centered inertial frame. Similarly, the Earth entry interface conditions targeted are listed in Table 4.6. These terminal constraints represent entry parameters that would allow for a safe crewed re-entry into the Earth's atmosphere[11, 24, 35, 55]. The epoch of the entry interface conditions is unconstrained. Convergence is achieved when the terminal and path constraints are met to within the tolerances listed in Table 3.13. Finally, for cases in which variable scaling is implemented, the scale values used are listed in Table 3.14.

Table 3.11: Initial Conditions

	Initial Value
Epoch	4-Apr-2024 15:30:00 TDT
Mass (kg)	20339.9 (total fuel = 8063.65 kg)
x (km)	-1236.7970783385588
y (km)	1268.1142350088496
z (km)	468.38317094160635
$\dot{x}$ (km/s)	0.0329108058365355
$\dot{y}$ (km/s)	0.589269803607714
$\dot{z}$ (km/s)	-1.528058717568413

Table 3.12: Terminal Constraints

Constraint	Value
Geodetic Altitude (km)	121.92
Longitude (deg)	175.6365
Geocentric Azimuth (deg)	49.3291
Geocentric Flight Path Angle (deg)	-5.86

Table 3.13: Constraint Tolerance Values

Constraint	Tolerance
Geodetic Altitude (km)	1e-4
Longitude (deg)	1e-4
Geocentric Azimuth (deg)	1e-4
Geocentric Flight Path Angle (deg)	1e-4

Table 3.14: Scale Values

	Value
Position (km)	1
Time (s)	3600
Δv (km/s)	0.001
Altitude (km)	1000
Longitude (rad)	π
Azimuth (rad)	π
Flight Path Angle (rad)	1

3.3.1 Finite Burn Example with Main Engine

The first case is representative of a nominal Earth return during which the maneuvers are performed by the vehicle's main engine. The initial guess file consists of 12 patch points (position, velocity, and time) taken from an optimized finite burn trajectory. It should be noted that the initial guess does not contain burn duration or direction information for the maneuvers; the algorithm must target these values on its own. The first patch point corresponds to a state on the initial lunar parking orbit. The interior patch points correspond to the states and epochs at each of the maneuver locations, both trans-Earth injection (TEI-1, 2, 3) and trajectory correction (TCM 1, 2, 3), and some additional waypoints along the trajectory. The final patch point in the initial guess is the state and epoch at the desired entry interface (EI).

For this case, both the impulsive targeter and the finite burn targeter are executed in order to find a feasible trajectory that satisfies the specified terminal constraints, while keeping the Δv sum of the individual maneuvers within the available budget. The finite burn algorithm is run both with and without variable scaling implemented. Table 5.3 compares the individual maneuvers, final Δv sum, and number of iterations required for convergence for the impulsive solution and the scaled and unscaled finite burn (FB) solutions. The burn parameters for each finite burn maneuver are given in Table 3.16 for the unscaled algorithm and in Table 3.17 for the scaled algorithm.

The individual maneuvers and total Δv sum for the finite burn targeter are fairly similar to the impulsive targeting results, which is to be expected

Table 3.15: Maneuver and Convergence Data

	Impulsive	Unscaled FB	Scaled FB
TEI-1 Δv (km/s)	0.7332	0.6900	0.6827
TEI-2 Δv (km/s)	0.2505	0.2598	0.2576
TEI-3 Δv (km/s)	0.4164	0.4133	0.4185
Total Δv (km/s)	1.4001	1.3630	1.3589
Iterations	5	7	6

Table 3.16: Burn Data - Unscaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)
TEI-1	381.0984	3975.542
TEI-2	123.4850	1288.171
TEI-3	176.8428	1844.789
Total	681.4262	7108.502

Table 3.17: Burn Data - Scaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)
TEI-1	377.5064	3914.0619
TEI-2	122.7568	1272.7671
TEI-3	179.4875	1860.9622
Total	679.7507	7047.7912

Table 3.18: Burn Data Using Auxiliary Engines - Unscaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	381.0984	3975.542	0.6900
TEI-2	917.1483	1347.401	0.2602
TEI-3	1408.5339	2069.305	0.4489
Total	2706.7806	7392.248	1.3991

given that the burn durations with the main engine are short enough for an impulsive assumption to be used. Interestingly, the Δv of the first maneuver is around 35 m/s lower with finite burn maneuvers incorporated than for the impulsive solution, resulting in a lower total Δv for those two trajectories. Since Δv values typically increase when transitioning from impulsive to finite thrust maneuvers[42], this indicates that the placement of the initial impulse in the startup arc may be non-ideal.

3.3.2 Finite Burn Example with Auxiliary Engines

For this example, a main engine failure is assumed to occur after TEI-1 and the auxiliary engines are used to perform the final two maneuvers. The trajectory is re-targeted from the first patch point after TEI-1, using the TEI-1 burn data and post-burn state from the previous example. All terminal and path constraints are the same. Burn data for each maneuver is listed in Tables 3.18 and 3.19, and the convergence data is given in Table 3.20.

The finite burn algorithm is able to meet the entry and cost constraints in the same number of iterations as in the previous example, where all three maneuvers are performed by the main engine. Interestingly, the total Δv for

Table 3.19: Burn Data Using Auxiliary Engines - Scaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	377.5064	3914.062	0.6827
TEI-2	939.5373	1380.293	0.2657
TEI-3	1344.6461	1975.446	0.4262
Total	2645.3467	7269.801	1.3746

Table 3.20: Convergence Data with Auxiliary Engines

	Unscaled	Scaled
Iterations	5	5

the scaled algorithm in this case is still lower than the impulsive solution. This will not always be the case; numerous feasible solutions can exist for any given set of patch points. Consider the results from the finite burn algorithm with and without scaling implemented: although the same algorithm is used, simply scaling the variables causes the targeter to converge on a different solution with a different total cost. In this particular case, the finite burn Level II process identified a lower cost solution than that determined with the impulsive targeter. In both cases, the total cost constraint is always enforced to ensure that the total cost is within the available fuel budget.

3.3.3 Finite Burn Example with Linear Steering

Linear steering is now applied to the auxiliary engine example, this time targeting the entire trajectory from the beginning with the assumption that the auxiliary engines will be used for the final two maneuvers. With variable scaling implemented in the Level I process, the algorithm converges

Table 3.21: Steering Rates, Ex. 1

Maneuver	$\dot{\mathbf{u}}$
TEI-1	$10^{-4}\left[\begin{array}{ccc} 0.1556 & -0.0134 & -0.0506 \end{array}\right]^T$
TEI-2	$10^{-5}\left[\begin{array}{ccc} -0.0495 & -0.2611 & -0.4879 \end{array}\right]^T$
TEI-3	$10^{-5}\left[\begin{array}{ccc} 0.5735 & 0.8048 & 0.0503 \end{array}\right]^T$

in 6 iterations. The converged final steering rates are listed in Table 3.21. For all three maneuvers, the burn is presumed, at first, to be inertially fixed, and so the initial guess for $\dot{\mathbf{u}}$ at each maneuver is $\mathbf{0}$.

Compared to the results above for the constant steering law, the burn data listed in Table 3.22 shows little change; this is unsurprising, given the small magnitude of the steering rates - the total change in the burn direction is less than 1° for each of the three maneuvers. Although the one extra iteration required for convergence is not significant in terms of performance, the marginally higher total Δv suggests that adding linear steering does not have any overall benefit for this particular problem. However, the sum of the Δv's for the last two maneuvers, the ones performed with the lower-thrust auxiliary engines, is reduced slightly with the implementation of linear steering. The higher total cost results primarily from the increased Δv of the first maneuver. Thus, as a second linear steering example, the steering law is now implemented only for the second and third maneuvers, while the thrust direction for the main engine maneuver is held inertially fixed. The resulting converged steering rates are given in Table 3.23, and the burn data for each maneuver is provided in Table 3.24. Now, the total Δv cost is lower than the

Table 3.22: Burn Data - Linear Steering Ex. 1

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	387.2992	4045.085	0.7083
TEI-2	915.1326	1344.440	0.2607
TEI-3	1349.7866	1982.998	0.4308
Total	2652.2184	7372.523	1.3998

Table 3.23: Steering Rates, Ex. 2

Maneuver	$\dot{\mathbf{u}}$
TEI-1	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
TEI-2	$10^{-5} \begin{bmatrix} -0.0484 & -0.1865 & -0.3728 \end{bmatrix}^T$
TEI-3	$10^{-5} \begin{bmatrix} 0.5342 & 0.8187 & 0.0870 \end{bmatrix}^T$

result for either the unscaled or the scaled algorithm in the previous section. The number of iterations increases to 7, but the Δv savings in this case offset the slightly slower convergence.

3.3.4 Finite Burn Example Over The Lunar Cycle

To further demonstrate the capabilities of the finite burn algorithm, return trajectories were generated over several days spanning the lunar cycle from February 1-28, 2024, 0:00:00 TDT, again using the auxiliary engines for

Table 3.24: Burn Data - Linear Steering Ex. 2

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	365.9789	3822.408	0.6649
TEI-2	953.9895	1401.525	0.2685
TEI-3	1359.3247	1997.011	0.4290
Total	2679.2931	7220.944	1.3624

the second and third TEI maneuvers. In this case, however, it is assumed that the use of the auxiliary engines is intentional and not the result of an unexpected main engine failure. Thus, all three burn maneuvers are targeted simultaneously by the algorithm. Running different cases over the entire lunar cycle allows for testing of the algorithm under varying Earth-Moon configurations. For these runs, only two entry constraints are targeted, geodetic altitude and geocentric flight path angle. The targeted values for these constraints are the same as those given in Table 4.6.

In this simulation, the input patch points to the finite burn algorithm come from a converged impulsive trajectory with the same initial point and entry targets. The total Δv of the impulsive trajectory for each case is 1.50 km/s. Tables 3.25 and 3.26 list results both without and with scaling, respectively, for days 1, 3, 6, 10, 13, 16, 19, 22, 25, and 28 of the lunar cycle.

Table 3.25: Unscaled Burn Data over the Lunar Cycle

Day	**TEI-1**		**TEI-2**		**TEI-3**		**Total Cost**	
	Δv (km/s)	Duration (s)	Δv	Duration	Δv	Duration	Total Δv	Iterations
1	0.6238	348.0345	0.4743	1651.2834	0.4019	1210.4498	1.5000	3
3	0.6992	385.6885	0.4402	1505.0542	0.3606	1079.8977	1.5000	5
6	0.6041	338.0300	0.5248	1823.7242	0.3709	1111.0038	1.4998	7
10	0.6045	338.2269	0.5781	1991.5292	0.3175	942.3371	1.5000	5
13	0.6168	344.5208	0.3691	1310.0636	0.5141	1577.7328	1.5000	4
16	0.6580	365.1850	0.3286	1158.8930	0.5135	1576.9396	1.5000	9
19	0.6014	336.6757	0.5866	2020.2097	0.3116	924.1053	1.4997	6
22	0.6978	384.9900	0.3844	1326.8775	0.4072	1233.2389	1.4894	7
25	0.6041	338.0202	0.6310	2155.7213	0.2650	779.7340	1.5000	5
28	0.6174	344.7791	0.4917	1710.5830	0.3910	1175.1849	1.5000	4

In the previous two examples, the impact of variable scaling is negligible. Here, however, several cases over the lunar cycle show a marked disparity

Table 3.26: Scaled Burn Data over the Lunar Cycle

Day	**TEI-1**		**TEI-2**		**TEI-3**		**Total Cost**	
	Δv (km/s)	Duration (s)	Δv	Duration	Δv	Duration	Total Δv	Iterations
1	0.6320	352.1669	0.4740	1646.1692	0.3939	1185.0331	1.4999	6
3	0.7159	393.8848	0.4303	1465.9122	0.3538	1058.6550	1.5000	5
6	0.6122	342.1556	0.5329	1844.8053	0.3548	1059.8574	1.4999	11
10	0.6045	338.2353	0.5765	1986.4955	0.3190	947.2553	1.5000	11
13	0.6154	343.7799	0.3698	1312.9152	0.5148	1580.1157	1.5000	6
16	0.7064	389.2130	0.3342	1159.8967	0.3965	1219.9719	1.4371	5
19	0.6017	336.8295	0.5992	2059.0593	0.2991	885.0120	1.5000	12
22	0.7172	394.5247	0.3867	1326.0707	0.3962	1194.1109	1.5001	9
25	0.6119	342.0196	0.6402	2178.5320	0.2480	727.6002	1.5001	20
28	0.6207	346.4521	0.4884	1698.1462	0.3910	1175.4618	1.5001	7

in convergence when scaling is applied. Most notably, the number of iterations required to converge for Day 25 jumps from 5 to 20 when scaling is introduced. Conversely, applying scaling to the case with the worst convergence unscaled, Day 16, reduces the number of iterations from 9 to 5 *and* decreases the total Δv noticeably. As was stated in the previous chapter, there are multiple feasible solutions for any given set of patch points; implementing variable scaling allows the algorithm to explore a different part of the solution space than it would otherwise, resulting in a different converged solution. Furthermore, some sets of scale values will produce better results for a given problem than others. The values used in these examples are chosen arbitrarily, and so performance of the scaled algorithm could potentially be improved if the scale values are determined in a more optimal manner. Several different scaling methods can be found in the literature[38]; however, such investigation is beyond the scope of the present study. While scaling clearly does not always provide a better solution than the one found by the unscaled algorithm, it adds a mea-

sure of flexibility to the finite burn algorithm. This is particularly useful for cases that will not converge when the unscaled targeter is applied, as shown in the following section.

3.3.5 Delayed Patch Point Simulations

Another test of the finite burn algorithm is whether or not it can converge on a feasible solution given a set of patch points that are not current. A set of patch points corresponding to a current or future departure time may not always be available, especially when ground communications are lost. The algorithm must therefore be able to converge even when the departure time listed in the input file has already passed. For this example, the input patch point file from the February 1 run in the previous section is used. However, the initial epoch is shifted so that it no longer matches that of the startup arc. The patch point times are updated to reflect this time shift, but the corresponding positions and velocities are not. This results in an initial guess trajectory with significant errors in the terminal constraint conditions; the algorithm must re-target the direction, duration, and ignition times of all three burns in order to correct these errors. As before, the auxiliary engines perform the TEI-2 and TEI-3 maneuvers and all three maneuvers are targeted simultaneously. To ensure that the characteristics of the initial lunar orbit remain the same, the patch points are converted to the MCI frame before the time delay is introduced. The initial time of the simulation is perturbed for 3, 6, 9, and then 12 hours. The initial entry constraint errors due to the time delay

Table 3.27: Initial Entry Constraint Errors

Delay (hr)	Altitude Error (km)	FPA Error (deg)
3	11040.163	1.6526
6	22415.148	4.3289
9	33857.773	7.3130
12	45332.406	10.3842

Table 3.28: Convergence with Delays

Delay (hr)	Unscaled	Scaled
3	10	17
6	DNC	23
9	DNC	29
12	DNC	35

*DNC - Did Not Converge

are given in Table 3.27, and the convergence data for the scaled and unscaled algorithm is listed in Table 3.28.

The benefits of variable scaling are most apparent in this example. Without scaling, the finite burn algorithm is only able to find a feasible solution for a 3 hour delay, and no more. After the scaling is implemented, however, the targeter is able to find a feasible solution that satisfies the entry constraints and fuel budget for even a 12 hour delay. As mentioned in the previous example, it is possible that the scaled algorithm performance could be even further improved with a more optimal set of scale values. These results underscore the adaptability of the two-level targeting structure; instead of trying to match a previously determined nominal trajectory, the algorithm explores the nearby solution space and is able to converge on a trajectory despite the poor quality

of the initial input.

3.4 Summary

This chapter presents the theoretical framework and numerical validation of a generalized linear targeting process that incorporates continuous control actuation. Several sample applications are demonstrated, including a pendulum with constant control, attitude slew targeting using reaction wheels, and orbital transfer with finite thrust maneuvers. The numerical solution process is as simple as that of an impulsive two-level targeting algorithm, though incorporating finite control capabilities does increase the complexity of the analytical gradients employed. This increased complexity naturally translates to an increase in computational overhead, in contrast to the impulsive targeting process. However, unlike the impulsive algorithm, the targeting process presented here is capable of addressing a wide range of control scenarios, which broadens the scope of applications for which this method may be utilized. This investigation successfully demonstrates that linear targeting algorithms for path planning and guidance need not be limited by impulsive assumptions. Furthermore, the algorithm presented serves as an alternative to nonlinear optimization methods which may not be feasible for autonomous applications.

Chapter 4

Initial Guess Determination

The goal of this segment of the investigation is to identify a systematic approach for the determination of accurate startup arcs for autonomous spacecraft path planning and guidance. This capability is of particular interest for on-demand onboard trajectory determination. Artificial potential function (APF) methods are common in the field of path planning, but the resulting control requirements are not always feasible in practice due to various hardware and mission constraints. Still, the general concept is useful in the determination of suitable startup arcs for autonomous algorithms that are capable of addressing these actuator constraints on demand. In this chapter, an approach to using APF methods for preliminary trajectory design is presented.

4.1 Potential Function Construction

There are numerous ways to construct the overall potential function for a given problem. Consider, for instance, a transfer between two intersecting orbits. One possible approach would be to accomplish the transfer by executing a maneuver at the point of intersection between the two paths. Subsequently,

the following candidate potential is identified,

$$P_{int} = k(\mathbf{r}_{int} - \mathbf{r}_0)^T(\mathbf{r}_{int} - \mathbf{r}_0), \tag{4.1}$$

where $\mathbf{r}_0$ is the current position, $\mathbf{r}_{int}$ is the point of intersection, and k is a user-defined weight. Designing a potential to achieve a transfer between non-intersecting orbits is more complicated. The approach selected in this study is based on a potential function control algorithm developed for microsatellites.[50] The construction of the artificial potential begins with the creation of a desired velocity field that is a function of the current state and some desired final target state or path constraint. The potential is then only explicitly a function of the error between the current and desired velocities,

$$P_{vel} = k(\mathbf{v}_t(\mathbf{r}_0, \mathbf{r}_{des}, \mathbf{v}_{des}) - \mathbf{v}_0)^T(\mathbf{v}_t(\mathbf{r}_0, \mathbf{r}_{des}, \mathbf{v}_{des}) - \mathbf{v}_0), \tag{4.2}$$

where $\mathbf{v}_0$ is the current velocity at position $\mathbf{r}_0$, $\mathbf{v}_t$ is the required transfer velocity (at the current position) as computed via the velocity field equations, and $\mathbf{r}_{des}$ and $\mathbf{v}_{des}$ are the desired (i.e. target) position and velocity. This simplifies to

$$P_{vel} = k\Delta \mathrm{v}^2 \tag{4.3}$$

which is simply the square of the maneuver cost scaled by the weighting term k.

Velocity fields can be constructed to impose any user-defined constraints on the trajectory, both along the path and at the endpoint. The calculation of the appropriate desired velocity, though, can be an extremely

complex endeavour. For example, of particular interest in this investigation is the design of a velocity field that guides the trajectory to a specified terminal state. If the transfer time is not specified, targeting a final state is equivalent to targeting a final orbit. Ideally, the computed desired velocity $\mathbf{v}_t$ at a given initial state minimizes the overall transfer Δv. However, for non-circular orbits, calculation of optimal transfer maneuvers, even between coplanar orbits, is an area of study unto itself.[5, 22, 47] Thus, optimization of the desired velocity field is beyond the scope of the present study. Specific details of the velocity field constructions used in this investigation will be discussed with the corresponding examples.

4.2 Maneuver Planning

Typically, path planning via the APF approach is accomplished by defining a virtual force that is equal to the sum of the gradients of each active potential with respect to the current position and velocity of the vehicle[51]. This virtual force yields the desired acceleration vector for the vehicle. This acceleration is then matched as closely as possible by the vehicle's actuators. Naturally, the resulting control history is continuous and unconstrained, a feature that is generally undesirable for the present application. Certainly, the approach is not intended for cases involving discrete control parameters, such as impulsive maneuvers more commonly employed in classical trajectory design. Reconciling APF methods with the use of discrete control has been considered for the problem of spacecraft formation flight[27], and the present

investigation adopts a similar approach. In this case, the time derivative of the potential, rather than the gradient, is selected to identify appropriate locations for impulsive maneuvers. Since the vehicle is constrained to stay on its current orbit until a maneuver is performed, changes in the potential P can result only from the propagation of the vehicle state. Instead of using actuation to drive P to some pre-specified value, the algorithm described here seeks the minimum value of the potential along the current path. If the current orbit does not intersect the desired orbit, then the potential is based on velocity error and the minimum corresponds to the point at which the desired maneuver will have the lowest Δv cost.

The overall potential field presented here consists of either the intersection potential defined in Equation 4.1 or the velocity potential in Equation 4.2. A graphical overview of the full APF maneuver planning algorithm is provided in Figure 4.1. Following initialization, the algorithm determines whether or not the vehicle is already on the desired path. If not, the next step is to check whether the vehicle has reached the user-specified maximum allowable number of maneuvers. If additional maneuvers are still permissible, then the current state is propagated forward in time until the necessary conditions for a minimum $\left(\frac{dP}{dt} = 0 \text{ and } \frac{d^2P}{dt^2} > 0\right)$ are met. At that point, a maneuver is performed to match the current velocity to the calculated desired velocity and next iteration of the algorithm begins.

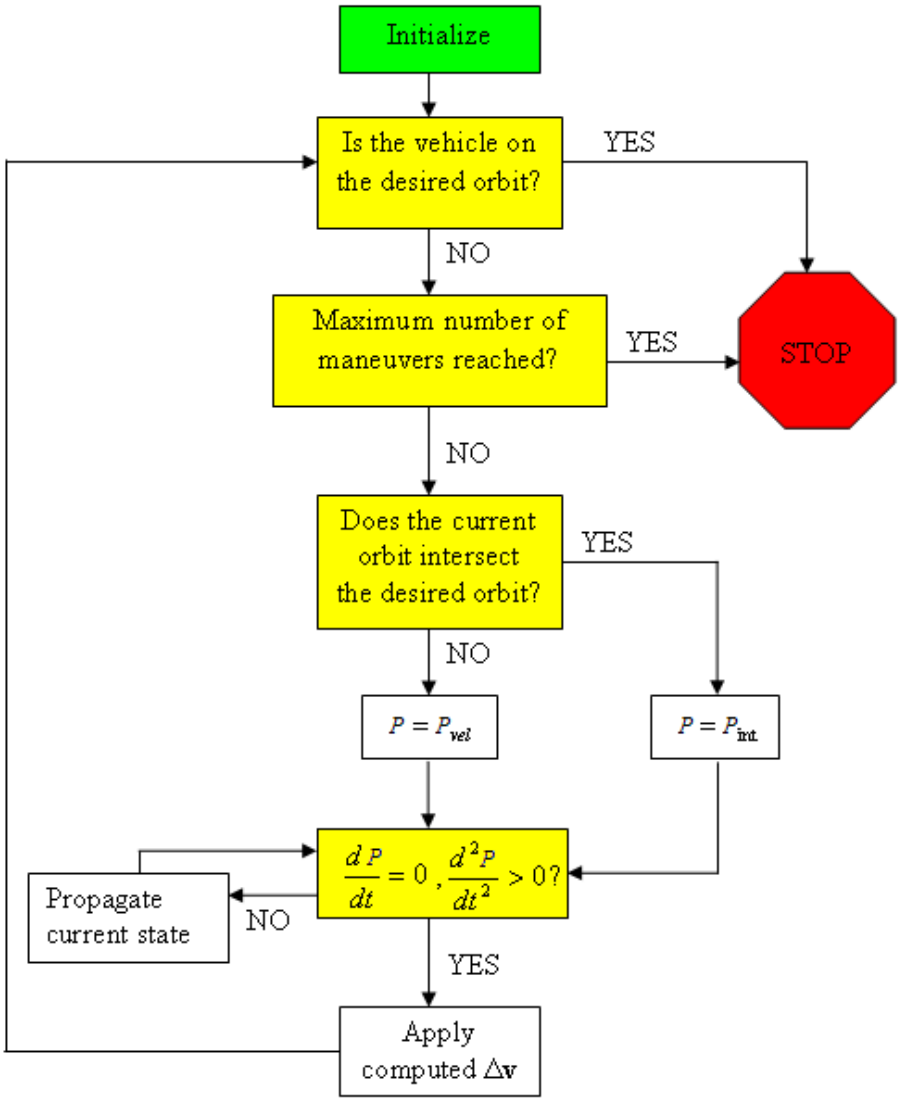


Figure 4.1: Overview of APF Maneuver Planning

4.3 Trajectory Design Examples

Initial testing of the APF trajectory design algorithm focuses on transfers in Earth orbit. The first example shown is a simple coplanar transfer, for which the analytical optimal solution is already known. This example provides a useful means for evaluating the performance of APF algorithm against an established benchmark. Subsequent examples build in complexity; transfer between coplanar elliptical orbits is examined, followed by an inclined two-body transfer and a lunar return example that incorporates third-body effects.

To determine the desired velocity field for the following examples, a target point on the final orbit is selected to be an apse of the transfer orbit. If a transfer proceeds from a lower orbit to a higher orbit, then the target point is defined to meet the apoapsis condition. If, instead, the transfer proceeds from a higher orbit to a lower orbit, a periapsis condition is imposed at the target point. Subsequently, the eccentricity vector of the transfer orbit is given by $\hat{\mathbf{e}}_t$,

$$\hat{\mathbf{e}}_t = -\hat{\mathbf{r}}_f, \tag{4.4}$$

where $\hat{\mathbf{r}}_f$ is a unit vector aligned with the Earth-to-target line. The magnitude of the target position vector can be determined from the target direction $\hat{\mathbf{r}}_f$ and the orbital parameters of the desired orbit,

$$r_f = \frac{p_d}{1 + \mathbf{e}_d^T \hat{\mathbf{r}}_f}, \tag{4.5}$$

where the general notation is as defined in Chapter 2 and the subscript d

denotes an element of the desired orbit. The eccentricity e_t is given by

$$e_t = \frac{r_f - r_0}{r_f - \mathbf{r}_0^T \hat{\mathbf{r}}_f}, \tag{4.6}$$

from which the parameter p_t can be calculated as

$$p_t = r_f \left(1 - e_t\right) = \frac{r_f \left(r_0 - \mathbf{r}_0^T \hat{\mathbf{r}}_f\right)}{\left(r_f - \mathbf{r}_0^T \hat{\mathbf{r}}_f\right)}. \tag{4.7}$$

Let $\hat{\mathbf{q}}$ be defined as a unit vector normal to both the angular velocity vector and periapse direction of an orbit, such that

$$\hat{\mathbf{q}} = \frac{\mathbf{h} \times \hat{\mathbf{e}}}{||\mathbf{h} \times \hat{\mathbf{e}}||}. \tag{4.8}$$

The desired transfer velocity at $\mathbf{r}_0$, $\mathbf{v}_{0t}$, can then be computed by

$$\mathbf{v}_{0t} = \sqrt{\frac{\mu \left(r_f - \mathbf{r}_0^T \hat{\mathbf{r}}_f\right)}{r_f \left(r_0 - \mathbf{r}_0^T \hat{\mathbf{r}}_f\right)}} \left[-(\hat{\mathbf{q}}_t^T \hat{\mathbf{r}}_0)\hat{\mathbf{e}}_t + \left(e_t + \hat{\mathbf{e}}_t^T \hat{\mathbf{r}}_0\right) \hat{\mathbf{q}}_t\right]. \tag{4.9}$$

Furthermore, the velocity at the terminal point of the transfer – the transfer orbit apoapsis – can also be computed:

$$\mathbf{v}_{ft} = \sqrt{\frac{\mu \left(r_f - \mathbf{r}_0^T \hat{\mathbf{r}}_f\right)}{r_f \left(r_0 - \mathbf{r}_0^T \hat{\mathbf{r}}_f\right)}} \left(e_t - 1\right) \hat{\mathbf{q}}_t. \tag{4.10}$$

Finally, the velocity of the desired orbit at point $\mathbf{r}_f$ is determined as

$$\mathbf{v}_{fd} = \sqrt{\frac{\mu}{p_d}} \left[-(\hat{\mathbf{q}}_d^T \hat{\mathbf{r}}_f)\hat{\mathbf{e}}_d + \left(e_d + \hat{\mathbf{e}}_d^T \hat{\mathbf{r}}_f\right) \hat{\mathbf{q}}_d\right], \tag{4.11}$$

which gives the required Δv at the intersection point $\mathbf{r}_f$ as $\Delta \mathrm{v}_f = |\mathbf{v}_{fd} - \mathbf{v}_{ft}|$.

4.3.1 Coplanar Transfer

Selection of the target point depends on the relative geometry of the initial and final orbits and can include numerous factors, particularly for more complex transfers involving third-body perturbations. For the purpose of this investigation, only the orbital planes are considered when designating $\hat{\mathbf{r}}_f$. If no plane change is required, the target point is chosen to be the position on the desired orbit that is directly opposite the current position, so that

$$\hat{\mathbf{r}}_f = -\frac{\mathbf{r}_0}{r_0}. \tag{4.12}$$

This choice of $\hat{\mathbf{r}}_f$ ensures that the transfer angle is always 180°. For the case of two circular orbits, this results in a Hohmann transfer. Also, from Equation 4.4, this means that the current position $\mathbf{r}_0$ is assumed to be the transfer periapsis.

Figure 4.2(a) shows the desired velocity field for transfer from a circular equatorial orbit to an elliptical equatorial orbit with perigee altitude of 800 km. The arrows indicate the direction of the desired velocity at various points along initial orbits with altitudes varying from 100 km to 780 km. Because the required velocity changes at both the start and end of the transfer can be computed a priori for a given $\mathbf{r}_0$, the potential in Equation 4.3 can be modified slightly to incorporate both the initial and terminal Δv's. The modified potential,

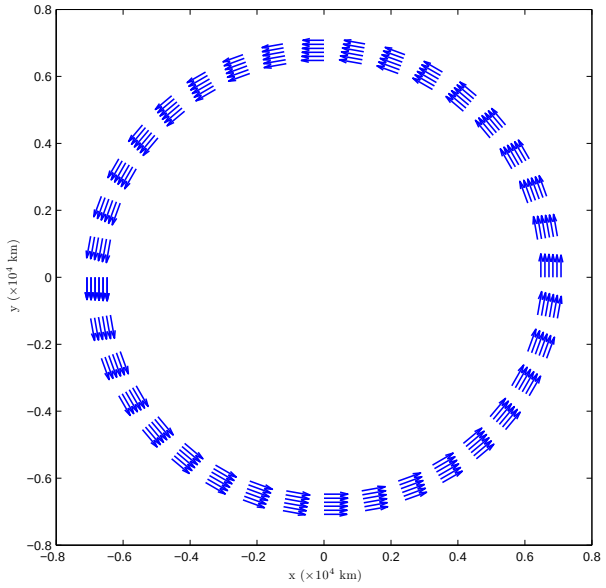
$$P_{vel} = k_0 \Delta \mathrm{v}_0^2 + k_f \Delta \mathrm{v}_f^2, \tag{4.13}$$

Table 4.1: Coplanar Transfer Initial and Target States

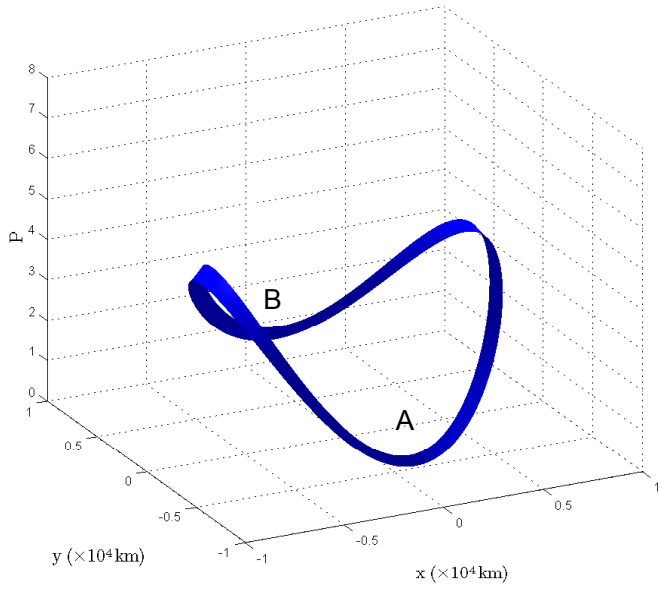
Parameter	Initial	Target
x (km)	-6478.145	0.000
y (km)	0.000	13330.841
z (km)	0.000	0.0000
v_x (km/s)	0.000	-4.575
v_y (km/s)	-7.844	0.000
v_z (km/s)	0.000	0.000

is given in Figure 4.2(b). From the figure, it is evident that two local minima exist for this potential. As mentioned previously, this is known to be a possible drawback to artificial potential function methods; however, this issue can be overcome through either judicious construction of the potential or incorporation of additional logic in the implementation of the algorithm, or a combination of both.

The transfer trajectory that ensues from the velocity field and potential described by Equations 4.9 and 4.13 is shown in Figure 4.3. The initial and target states for this transfer are listed in Table 4.1. An analytical optimal result exists for the case of coplanar, coaxial orbits in the form of a doubly cotangential transfer through an angle of 180°, terminating (or originating, for transfers from higher to lower orbit) at the higher apogee of the two orbits[21]. As Figure 4.3 shows, the APF method matches this result, which has a total Δv of 1.409 km/s. Note, though, that the initial vehicle state is prior to the analytical optimal impulse point. Had the initial state been on the other side of this point, the algorithm as-is would have converged to the other minimum directly opposite this impulse point (as shown in Figure 4.2(b)). This



(a) Velocity Field



(b) Resulting Potential

Figure 4.2: Coplanar Velocity Field and Potential Function

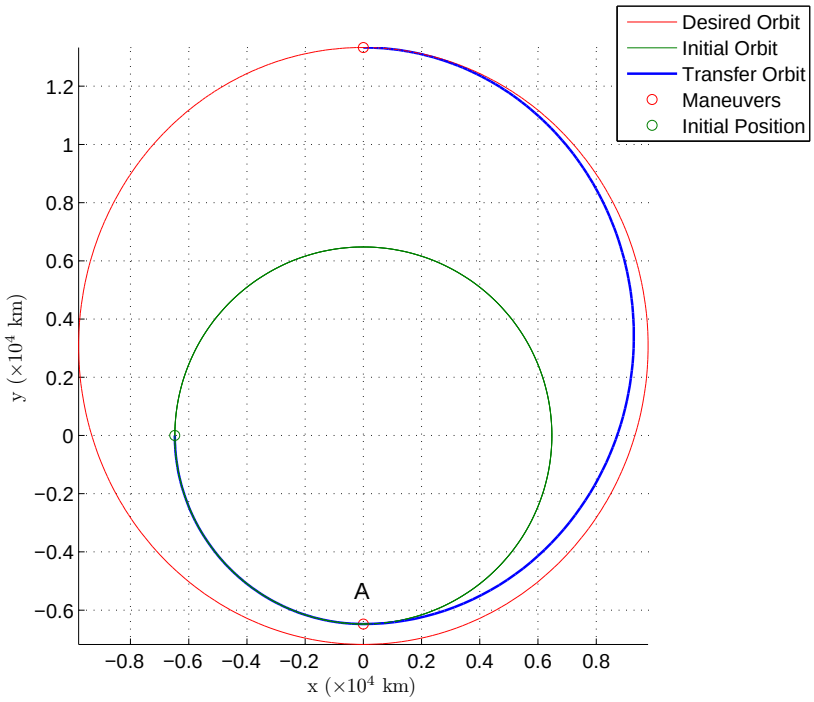


Figure 4.3: Coplanar Transfer

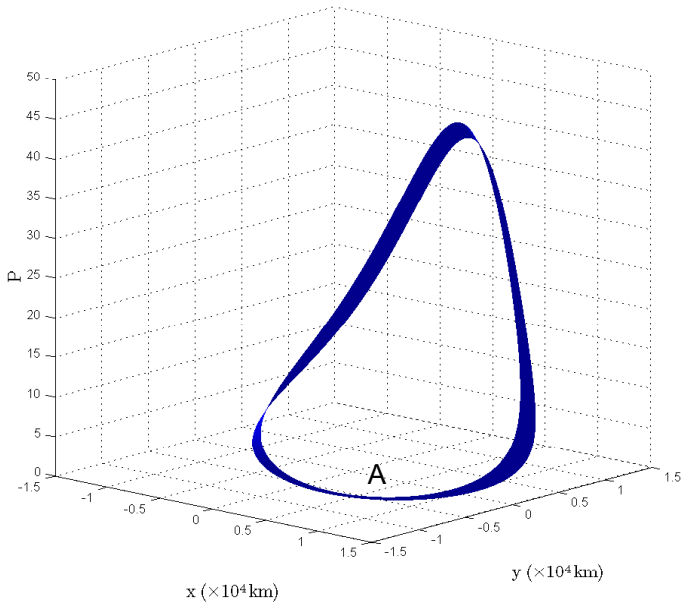
transfer requires approximately an additional 28 m/s of Δv, but does offer a significantly shorter transfer time. Depending on the requirements of the problem, the desired velocity field can, if needed, be computed in such a way that it forces the optimal result for this problem. This is explored further in the next example. Although the problem addressed here is fairly trivial, the result demonstrates that the APF algorithm is capable of performing on par with analytical methods, when they are available.

For the second coplanar example, the vehicle begins in an elliptical orbit, rather than a circular one. The initial orbit's apsidal line is rotated to be non-coaxial with the terminal orbit, so that there is no simple analytical solution to determine the optimal transfer. Also, the eccentricity and parameter of the desired orbit are both increased to prevent an intersection between the two orbits. Table 4.2 lists the initial and desired terminal states. The target point on the desired orbit is selected differently for this example. From the analytical result for directly coaxial orbits, i.e. that the optimal transfer has an endpoint at the highest apoapsis[21], $\mathbf{r}_f$ is assumed to be fixed at the apogee of the final orbit,

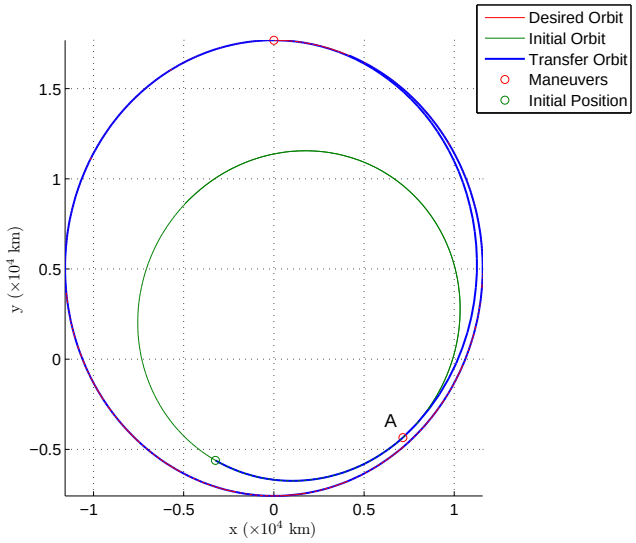
$$\mathbf{r}_f = \mathbf{r}_{d,apogee}. \tag{4.14}$$

This designation of the target point could also be used in the previous example to avoid the local minimum of the potential that occurs 180° from the true optimal solution. Computation of the desired velocity field follows the same procedure described in the previous section.

Figure 4.4 shows the subsequent potential function surface (4.4(a))and



(a) Potential



(b) Resulting Transfer

Figure 4.4: Potential Function and Transfer for Elliptical Initial Orbit

Table 4.2: Coplanar Transfer Initial and Target States

Parameter	Initial	Target
x (km)	-3239.073	0.000
y (km)	-5610.238	17682.338
z (km)	0.000	0.0000
v_x (km/s)	7.745	-3.678
v_y (km/s)	-4.472	0.000
v_z (km/s)	0.000	0.000

transfer arc (4.4(b)). Here, the minimum of the potential does not coincide with the either apsidal line, nor are there any local minima. The lowest point of the potential surface occurs instead at a true anomaly of close to 90° on the initial orbit, as indicated by the location of the first maneuver in Figure 4.4(b). The Δv of the resulting transfer is just 0.7955 km/s, compared to 1.3728 km/s for the 180° transfer to the terminal orbit apoapsis. For situations like this one, in which a simple analytical solution is not known, the artificial potential function method is a fast and efficient approach to finding a practical transfer.

Other factors, in addition to Δv cost, may be incorporated into the potential function as well. One such parameter is the time-of-flight along the transfer arc. Consider again the first example in this section, the transfer from a circular orbit to a higher altitude elliptical orbit. Now, it is desired to limit both the Δv and the time-of-flight when determining a suitable transfer arc. Thus, the potential function from Equation 4.13 is augmented to include a time-of-flight term,

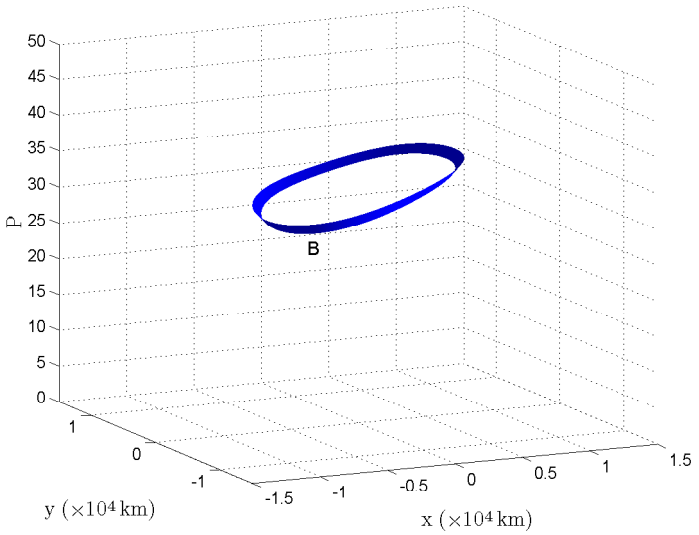
$$P = k_0 \Delta \mathrm{v}_0^2 + k_f \Delta \mathrm{v}_f^2 + k_T \Delta t_{TOF}, \tag{4.15}$$

where the time-of-flight Δt_{TOF} is computed using two-body analysis. The target point $\mathbf{r}_f$ and the transfer velocities are calculated as described in the initial example. Figure 4.5 shows the potential surface and transfer orbit that proceed from the definition in Equation 4.15, using the same initial and target conditions listed in Table 4.1. The addition of the time-of-flight term eliminates the minimum associated with the previous solution, leaving only one position along the initial orbit where the conditions for executing a maneuver are satisfied. The resulting transfer is characterized by both a low Δv cost and a small transfer time.

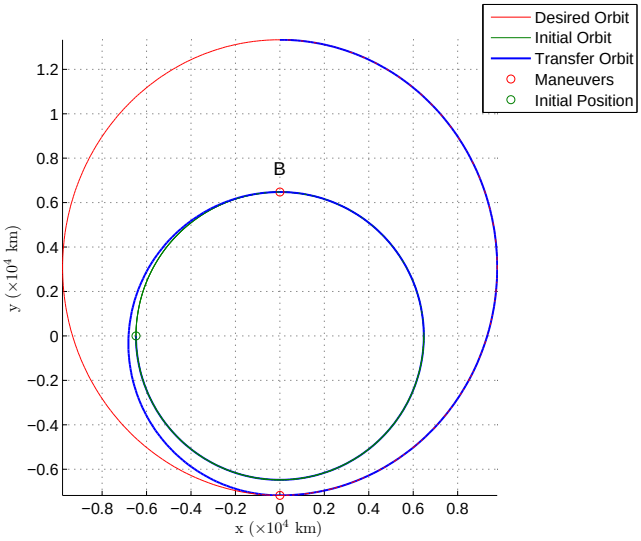
Similar adjustments to the potential function may be made for a variety of transfer orbit characteristics. Parameters of interest might include the terminal transfer time (rather than time-of-flight), terminal flight-path angle, or altitude of closest approach to the central body, among many others. The flexibility to incorporate any or all of these constraints, and to assign varying degrees of importance to each one, is a significant advantage to the APF design approach.

4.3.2 Inclined Transfer

If the initial and desired orbits are non-intersecting and not coplanar, a two or three-impulse nodal transfer is assumed. Thus, the target point is the point where the desired orbit intersects the plane of the initial orbit. This target vector is perpendicular to the angular velocity vectors of both orbits,



(a) Potential



(b) Resulting Transfer

Figure 4.5: Potential Function and Transfer with Time-of-Flight Term

so its direction is given by

$$\hat{\mathbf{r}}_f = \pm \frac{\mathbf{h}_0 \times \mathbf{h}_{des}}{||\mathbf{h}_0 \times \mathbf{h}_{des}||}. \tag{4.16}$$

Equation 4.16 yields two results for the direction of intersection, although both answers may not always be valid depending on the geometry of the two orbits. If two legitimate intersection points do exist, the selected target point will depend on the relative alignment of the orbit apsidal lines and the nodal line.

4.3.2.1 2-Maneuver Transfer

Several previous studies have considered the case of optimal two-impulse transfer between arbitrary elliptical orbits, and some useful analytical results have been determined from this research[21, 49, 56]. Of particular interest for this investigation is the fact that, for a nodal transfer, the radial components of the initial and final velocities at the impulse points affect the total transfer Δv only through their algebraic sum; thus, they have no effect on the optimal transfer plane orientation[49]. Furthermore, the optimal transfer between the nodal points is equivalent to the optimal transfer between two coaxial pseudo-orbits that lie in the plane of the actual initial and final orbits. The apsidal points of these pseudo-orbits coincide with the intersections of the actual orbits with the nodal line[56]. If these equivalent pseudo-orbits are directly coaxial–that is, their periapse directions are the same–or one orbit is circular, then the optimal transfer occurs between the nodal point having the greatest radial distance from the central body and the node with the smaller radial distance on the other orbit, i.e. the perigee of the pseudo-orbit. For

inversely coaxial pseudo-orbits, with opposite periapse directions, the optimal transfer can either be between the two higher altitude nodes or the two lower altitude nodes[21]. Experimental results[49] indicate that, for a given total plane change angle θ_T, the lower Δv transfer orbit is the one with the greater distance ratio $\frac{r_f}{r_0}$. Therefore, in this study, the distance ratio will be the criterion for selecting the target nodal point in order to determine the desired velocity field.

With the target point $\mathbf{r}_f$ chosen, the desired velocity can be calculated according to the formulas presented in the previous section. For a general point $\mathbf{r}_0$ that does *not* fall on the nodal line, it is assumed that the transfer orbit is in the plane of the initial orbit, and the desired velocity is thus given by Equation 4.9. If, however, $\mathbf{r}_0$ is on the nodal line, then it is preferable to accomplish a small percentage of the plane change at the initial maneuver. As discussed in Chapter 2, there is no known analytical solution to find the optimal plane change split over the two maneuvers. However, if the cost function to be minimized is instead defined as the sum of the squares of the Δv's (i.e. the potential in Equation 4.13), the optimal initial plane change is given by[54]

$$\tan\theta_1 = \frac{\sin\theta_T}{\left(\frac{r_f}{r_0}\right)^{(3/2)} + \cos\theta_T}. \tag{4.17}$$

This equation for the plane change split gives a result that is within 0.5% of the optimal. A very similar formula, developed for this study, gives the initial plane change as

$$\tan\theta_1 = \frac{\sin\theta_T}{\left(\frac{r_f}{r_0}\right)^{2} + \cos\theta_T}. \tag{4.18}$$

A comparison of this equation with the formula in Equation 4.17 is shown in Figure 4.6 for transfer between circular orbits. The Δv resulting from the plane change in Equation 4.18, marked by the dashed red line, closely follows the Δv curve produced by Equation 4.17 and in fact performs better at lower ratios of r_f to r_0. Additionally, this formulation transitions more smoothly to the three-maneuver plane change split, as discussed in the next section.

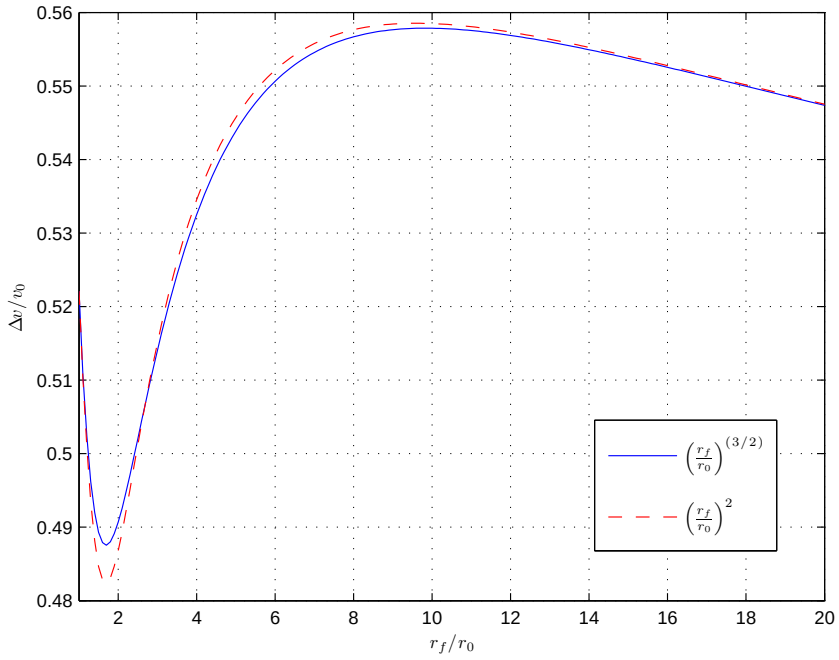


Figure 4.6: Comparison of Plane Change Division Formulas

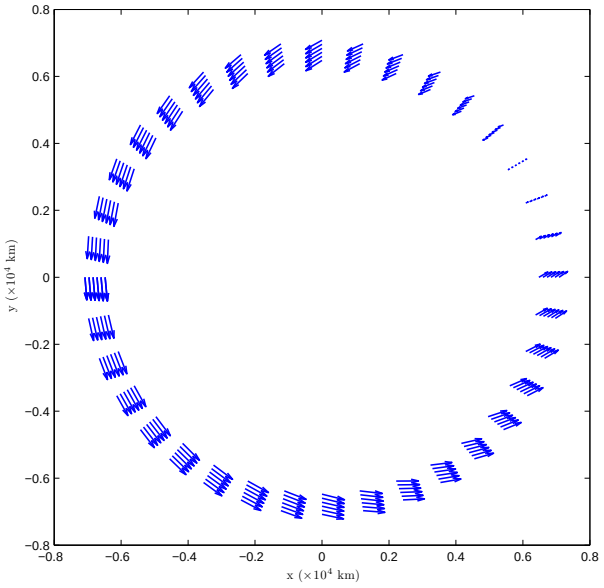
An example velocity field generated by Equations 4.9 and 4.18, and its associated potential, are shown in Figures 4.7(a) and 4.7(b), respectively. The initial and target states for this example are given in Table 4.3. Like the potential in Figure 4.4(a), from the second coplanar example, this surface displays

Table 4.3: Inclined Transfer Initial and Target States

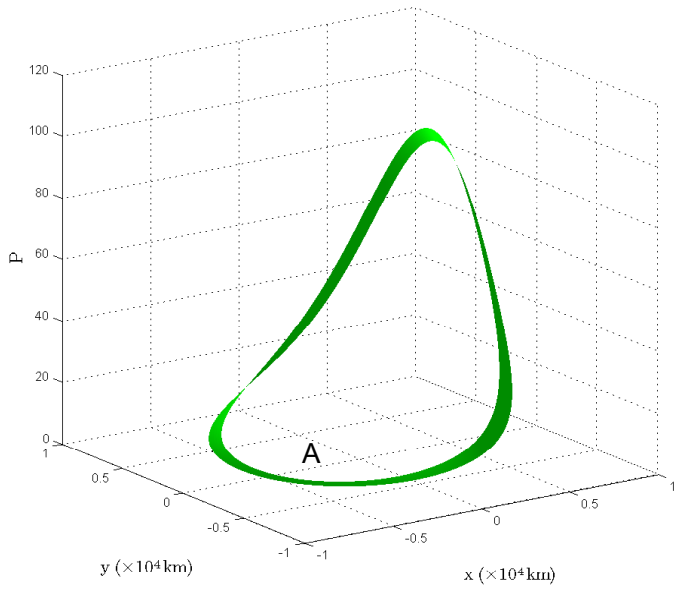
Parameter	Initial	Target
x (km)	-5610.238	201.755
y (km)	0.000	13128.315
z (km)	3239.073	2306.067
v_x (km/s)	0.000	-4.558
v_y (km/s)	-7.844	0.000
v_z (km/s)	0.000	0.399

no unwanted local minima. Here, the lowest point on the surface coincides with the point on the initial orbit that is exactly opposite $\mathbf{r}_f$, corresponding to a 180° nodal transfer.

The non-coplanar transfer, shown in Figure 4.8, requires a much greater Δv than the coplanar example due to the 26.8° plane change. This transfer has a total cost of 3.30 km/s, compared to 1.37 km/s for the coplanar example. Using a Lambert targeting algorithm[10] to achieve this same transfer, for the same total time, results in a Δv of 6.84 km/s. In fact, even the Lambert solution from the initial point to the terminal point of the transfer arc only, using the same transfer time, yields a slightly greater Δv – 3.47 km/s – simply because the Lambert algorithm chooses a less favorable transfer orbit plane. Because of the high cost incurred by changing the orbital plane, it is often advantageous to add a third maneuver, so that the majority of the plane change is effected at a high radial distance from the central body[39]. This is explored further in the following section.



(a) Velocity Field



(b) Resulting Potential

Figure 4.7: Non-Coplanar Velocity Field and Potential Function

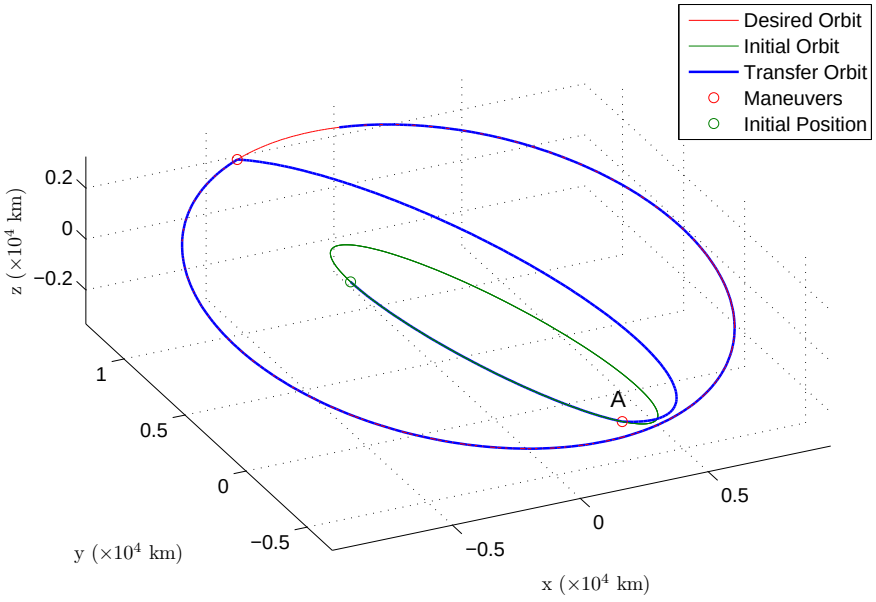


Figure 4.8: Non-Coplanar Transfer

4.3.2.2 3-Maneuver Transfer

As shown in Chapter 2, adding a third maneuver can greatly reduce the required Δv when a significant plane change is required. Between circular orbits, the optimal solution is a bi-elliptical transfer, with a certain percentage of the plane change incorporated into each maneuver. It is again assumed that the target points for each maneuver are on the nodal line defined by Equation 4.16, but now there will be an intermediate point, $\mathbf{r}_b$, that does not lie on either the initial orbit or the final orbit. The preferred radius of the intermediate point r_b depends on several factors, including the ratio $\frac{r_f}{r_0}$, the total plane change θ_T, and any time constraints on the transfer. For this study, r_b is chosen such that it minimizes the cost function

$$J = \frac{\gamma_1 \left(1 - \hat{\mathbf{h}}_d^T \hat{\mathbf{h}}_0\right) r_0}{r_b} + \gamma_2 \left[\frac{p_d}{r_b} - \left(1 + \mathbf{e}_d^T \hat{\mathbf{r}}_b\right)\right]^2, \tag{4.19}$$

where γ_2 is a user-defined constant tuning parameter and γ_1 varies with the current orbit. The first term of J seeks to increase r_b as θ_T increases (i.e. as $\hat{\mathbf{h}}_d^T \hat{\mathbf{h}}_0 = \cos\theta_T$ decreases), while the second term adds a penalty for straying too far from the radius that the desired orbit has along the direction of $\hat{\mathbf{r}}_b$. The desired value of r_b is found by setting the partial derivative of J with respect to r_b equal to 0,

$$\frac{\partial J}{\partial r_b} = -\frac{\gamma_1 \left(1 - \hat{\mathbf{h}}_d^T \hat{\mathbf{h}}_0\right) r_0}{r_b^2} - \frac{2\gamma_2 p_d^2}{r_b^3} + \frac{2\gamma_2 p_d}{r_b^2}\left(1 + \mathbf{e}_d^T \hat{\mathbf{r}}_b\right) = 0, \tag{4.20}$$

which gives the result

$$r_b = \frac{2\gamma_2 p_d^2}{2\gamma_2 p_d \left(1 + \mathbf{e}_d^T \hat{\mathbf{r}}_b\right) - \gamma_1 \left(1 - \hat{\mathbf{h}}_d^T \hat{\mathbf{h}}_0\right) r_0}. \tag{4.21}$$

When the initial and final orbits are coplanar, $1-\hat{\mathbf{h}}_d^T\hat{\mathbf{h}}_0 = 0$ and r_b corresponds to the final orbit radius along the direction $\hat{\mathbf{r}}_b$. The value of the parameter γ_1 depends on the vehicle's present orbit; if its apoapsis is high, γ_1 is smaller, but if the apoapsis altitude is low, γ_1 increases to force a greater r_b for the subsequent maneuver. For this investigation, a hyperbolic secant function is used to determine γ_1. The hyperbolic secant, $\text{sech}(x)$, takes a value of 1 when x is 0, and goes to 0 as x goes to infinity, as shown in Figure 4.9. The equation for γ_1 is given by

$$\gamma_1 = \lambda_1 \text{sech}\left(\frac{\lambda_2\left(r_{0ap} - r_{0min}\right)}{r_{0min}}\right), \tag{4.22}$$

where r_{0ap} is the apoapse radius of the current orbit, r_{0min} in the minimum radius of the starting orbit prior to any transfer maneuvers, and λ_1 and λ_2 are user-defined constants.

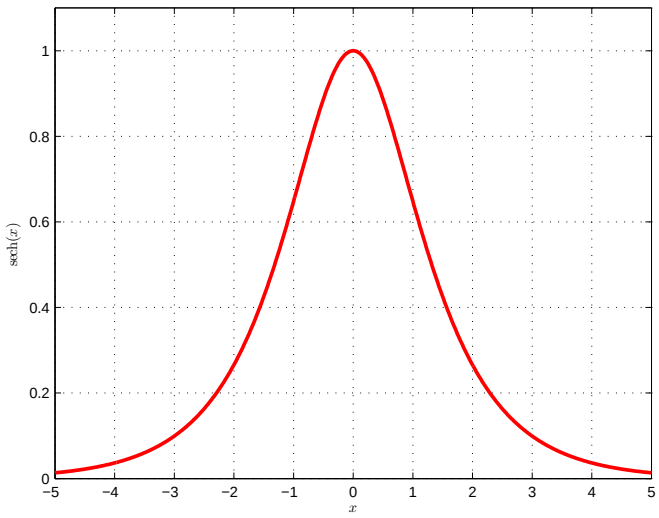


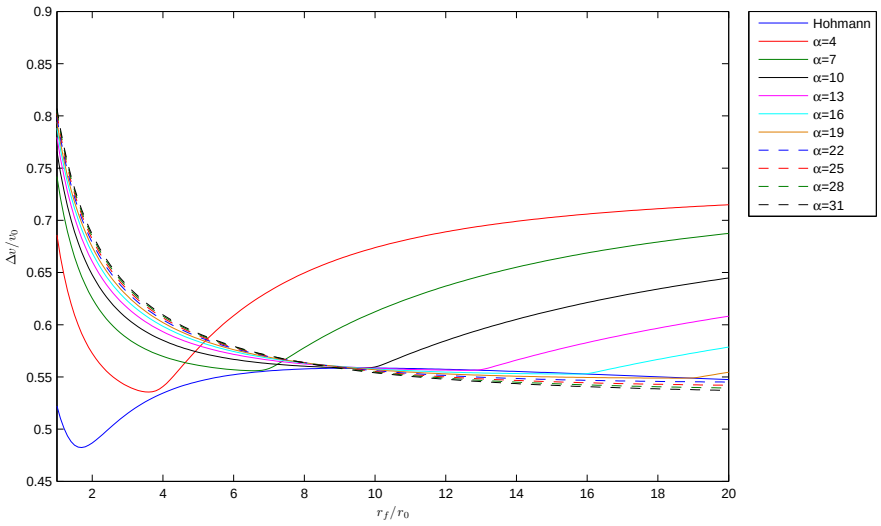
Figure 4.9: Hyperbolic Secant Function

Having determined the target terminal point, it is next necessary to find the orientation of the two transfer planes. As with the two-impulse transfer, there is no analytical solution for the optimal division of the total plane change, but the approach presented in the previous section for achieving a near-optimal split can be extended to provide a formula for the three-impulse sequence. A good approximation for the plane change at each maneuver is found to be

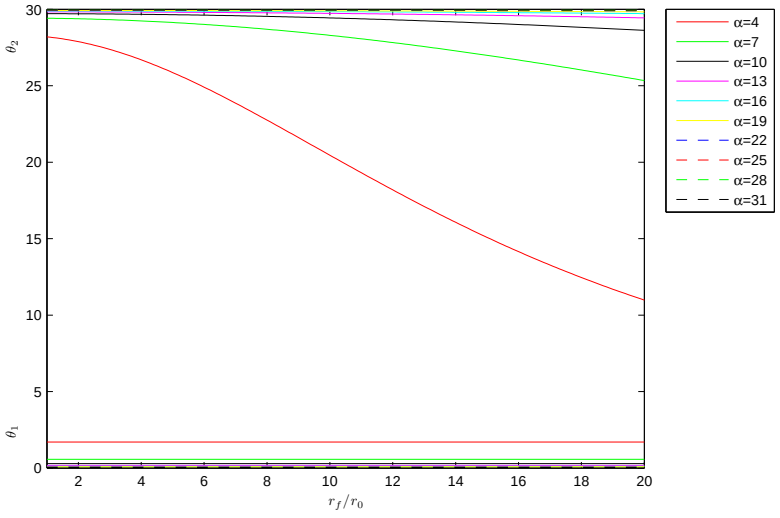
$$\tan\theta_j = \frac{(r_j/r_0)^2 \sin(\theta_T - \theta_{j-1})}{(r_{j+1}/r_j)^2 + (r_j/r_0)^2 \cos(\theta_T - \theta_{j-1})},\ j = 1, 2 \qquad (4.23)$$

with the change at the final maneuver given by $\theta_3 = \theta_T - (\theta_1 + \theta_2)$. In the case of a circular initial orbit, if the first impulse goes to 0 such that $\theta_1 = 0$ and $r_1 = r_2 = r_0$, Equation 4.23 reduces to the two-impulse θ_j division from Equation 4.18. For a total plane change of $\theta_T = 30°$, this approximation produces the Δv and θ_j curves shown in Figure 4.10. Comparing these curves to the optimal results shown in Figure 2.9, it is evident that, although the shape of the θ_j curves do not follow those of the optimal plane change split particularly closely, the transfer Δv plots are quite similar. This trend also holds for all other values of θ_T. With the target points and transfer plane orientations set, computation of the desired velocity field and the potential follow as described previously.

The following transfer example utilizes the three-maneuver formulation described above, with tuning parameters $\gamma_2 = 1$, $\lambda_1 = 3$, and $\lambda_2 = 60$. The initial orbit is the same as in the inclined two-maneuver example, but the final orbit has been rotated further so that the total required plane change is 65.9°.



(a) Δv



(b) Plane Change Split

Figure 4.10: Hohmann vs. Bi-elliptic Inclined Transfer, 30^o

Table 4.4: Inclined Transfer Initial and Target States

Parameter	Initial	Target
x (km)	-5610.238	2952.760
y (km)	0.000	5319.912
z (km)	3239.073	11019.849
v_x (km/s)	0.000	-4.548
v_y (km/s)	-7.844	0.000
v_z (km/s)	0.000	1.219

The beginning and target states are listed in Table 4.4, and the resulting transfer orbit is displayed in Figure 4.11. The total Δv for this transfer is 6.97 km/s, which is significantly less costly than the 7.95 km/s required for the two-maneuver sequence for this same problem. Due to the high (33588.4 km) radius of intermediate point, however, the transfer time is 6.24 hours longer than the two-impulse transfer. Depending on the constraints of the problem, this may or may not be an acceptable trade for the Δv savings. In addition, smaller plane changes such as the previous inclined transfer example are more efficiently achieved (in both fuel and transfer time) with a two-maneuver transfer; for the 26.8° orientation change, the total cost with three maneuvers is 4.15 km/s, compared to 3.30 km/s for two maneuvers. The choice of a two-impulse vs. a three-impulse sequence is problem specific, and at this point no guaranteed global selection criteria have been found. Currently, this decision is made by the user, but future research will investigate methods for automating the choice of sequence.

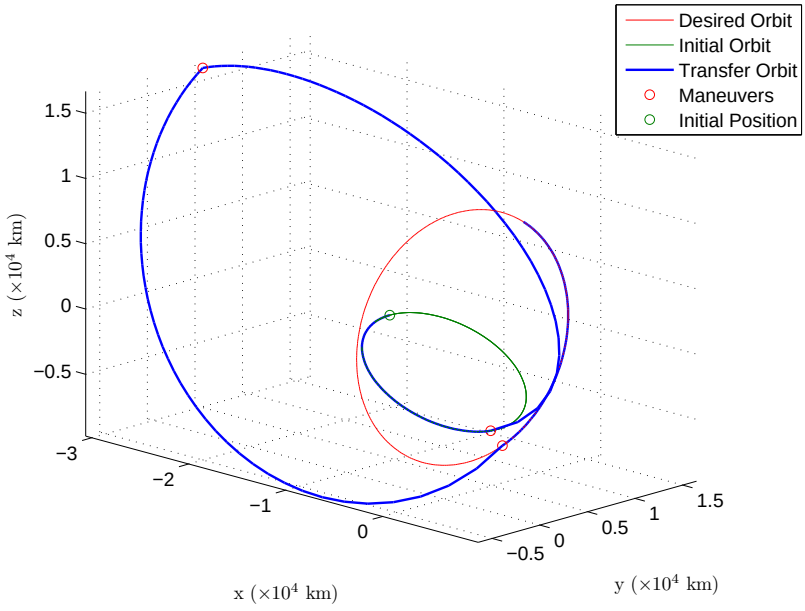


Figure 4.11: Non-Coplanar Transfer

4.3.3 Lunar Transfer

In order for the APF method to be a useful general tool for identifying startup arcs, it must be able to produce valid results for more complex cases, such as cislunar trajectories. In the example below, the velocity field construction described previously is employed in the APF algorithm to design a return trajectory from a low-lunar parking orbit. A two-impulse transfer sequence is assumed for any plane changes required. Initial conditions for the parking orbit are listed in Table 4.5. To determine an appropriate target orbit, i.e. one that will return the spacecraft to some acceptable Earth entry interface, a set of entry constraints is chosen along with an estimated arrival time. Currently, these entry conditions are assumed to come from a database or prior

optimization. The development of an autonomous method for determining these constraints is the subject of future research. These constraints are used to compute the entry state (position and velocity), which is then propagated backwards in time until reaches the lunar sphere of influence. The position and velocity at the end of the propagation are input as the desired final state for the APF algorithm.

The entry interface constraints and arrival time for this example are given in Table 4.6. It should be noted that the timing of the vehicle's arrival at the target state is now significant, as the entry parameters will depend on the relative configuration of the Earth and Moon. Because the current APF method determines maneuvers independent of time, there may exist a mismatch between the assumed time of arrival at the desired state, determined via the backward propagation, and the actual time of arrival of the APF-determined departure trajectory. This is rectified by adjusting the vehicle's initial departure time through a simple offset targeting algorithm. At each iteration, the arrival time error is computed and that value is subtracted from the current departure time. This method typically reduces the arrival time error to within 1 second in 3 or 4 iterations. If it is necessary for the initial time to be strictly enforced, this can be accomplished through the targeting algorithm; this will be explored in further detail in Chapter 5.

The trajectory generated by the APF algorithm is shown in Figure 4.12. Using the offset method to correct the departure timing results in an initial time that is 15.4 hours earlier than the original start time for the problem.

Table 4.5: Initial Conditions

Epoch	2-Aug-2018 1:53:38.4 TDT
x (km)	-1834.7155
y (km)	-66.2361
z (km)	-73.9653
v_x (km/s)	-0.0864
v_y (km/s)	0.8139
v_z (km/s)	1.4136

Table 4.6: Estimated Arrival Conditions

Epoch	7-Aug-2018 00:52:07.9 TDT
Geocentric Altitude (km)	121.92
Longitude (deg)	-134.5456
Geocentric Latitude	-19.20410
Geocentric Azimuth (deg)	13.9960
Geocentric Flight Path Angle (deg)	-6.0300

This arc has a total Δv of 1.6886 km/s, which is slightly high for a lunar return, but this is an issue than can be addressed to some extent by a targeting or optimization algorithm. Ideally, though, the cost of the startup arc should be as low as possible before being passed on to the targeting routine, so it is beneficial to examine ways in which the cost of the APF trajectory can be reduced.

Phasing is extremely important when designing trajectories in multi-body systems. Using two-body approximations to construct the artificial potential field neglects the effects of phasing, a limitation that can have a detrimental effect on the performance of the APF algorithm. Consider the return example described previously. If the departure and entry times for this trans-

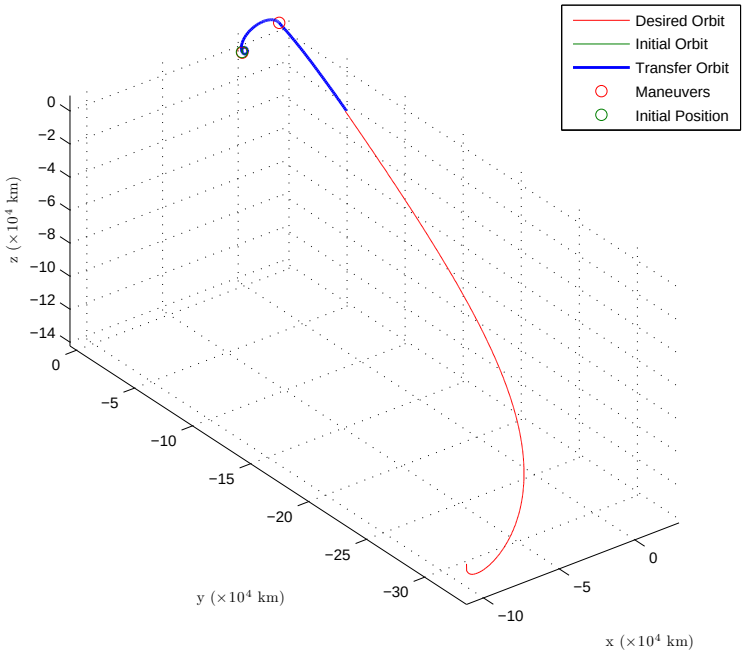
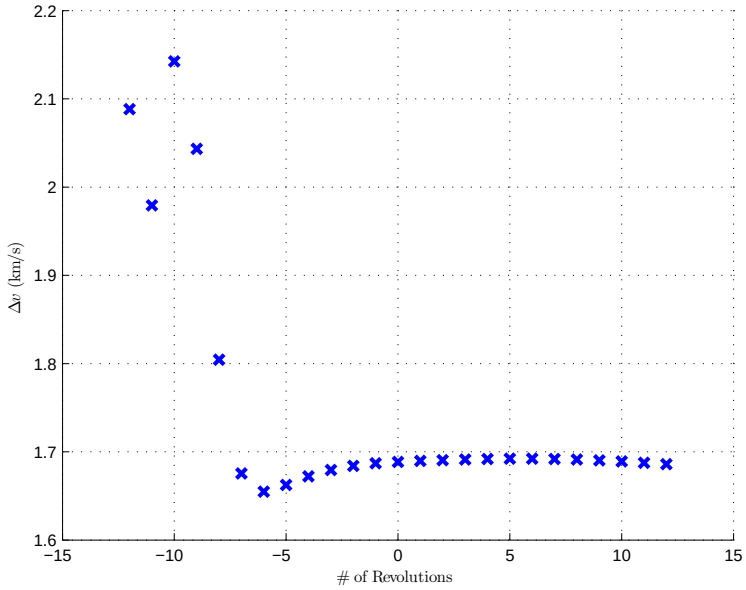


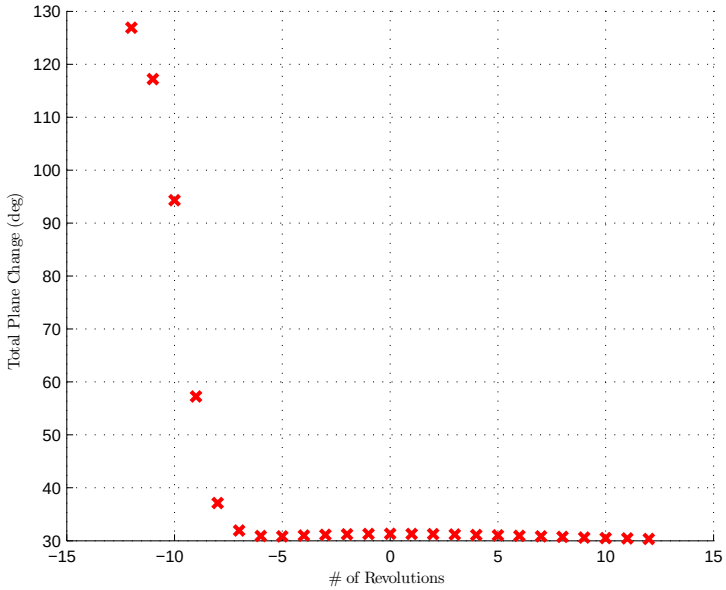
Figure 4.12: APF Lunar Return (MCI), 1.6886 km/s

fer were shifted, the change in the Earth-Moon configuration could cause a noticeable difference in the trajectory produced by the APF algorithm. To demonstrate this timing effect, the transfer above is subjected to time shifts over a range of ± 12 revolutions of the initial orbit. These arcs have the same initial conditions and entry parameters as those listed in Tables 4.5 and 4.6, except that the epochs for departure and arrival are shifted by n revolutions. The costs of the resulting transfers are plotted against n in Figure 4.13(a), and the total plane change associated with this geometry is given in Figure 4.13(b). There is a definite, though not exact, correlation between the plane change required and the total Δv cost, which is to be expected. Changes in orbit orientation, as discussed previously, have a considerable impact on the transfer cost, but there are also many other influencing factors, particularly for complex problems such as cislunar trajectories. It is clear from Figure 4.13(a) that improvements in cost can be achieved for this transfer simply by departing a few revolutions earlier, if possible; the total Δv drops from 1.6886 km/s to 1.6548 km/s, a reduction of over 30 m/s, in the course of 6 revolutions. Conversely, a poor choice of transfer epoch can have a disastrous effect on cost, as evidenced by the 2.1425 km/s trajectory that results from leaving 10 revolutions earlier. Future iterations of the APF algorithm must be capable of taking these phasing effects into account in order to be truly effective as a general trajectory design tool.

Another important factor in more complex trajectory design is the set of desired terminal conditions. For lunar returns, this corresponds to the entry



(a) Δv



(b) Plane Change

Figure 4.13: Δv and θ_T of Time-shifted Return Trajectories

Table 4.7: Modified Arrival Conditions

Epoch	7-Aug-2018 00:53:09.7 TDT
Geocentric Azimuth (deg)	44.6262
Geocentric Flight Path Angle (deg)	-5.8600

interface parameters and entry epoch listed in Table 4.6. Even slight modifications to these constraints can have a noticeable effect on the return trajectory. For example, consider the two-maneuver return generated previously. With the same entry interface conditions, switching to a three-maneuver sequence has little effect on the overall cost of the return, other than significantly increasing the time-of-flight. However, if the entry azimuth, flight path angle, and epoch are adjusted to the values in Table 4.7, the APF algorithm with a three-maneuver sequence produces the initial guess displayed in Figure 4.14. This trajectory has a total Δv of 1.4928 km/s (1.7216 km/s with the two-maneuver sequence), and it requires only a +2.6 hour initial time shift to match the time at the end of the backward propagation. Clearly, selection of the terminal conditions plays an significant role in the overall trajectory design, and should be examined further as part of future research efforts in developing the APF design algorithm.

4.4 Summary

This chapter examines the use of artificial potential function (APF) methods as a means of identifying suitable startup arcs for targeting and optimization algorithms. The primary objectives of the study are to evaluate

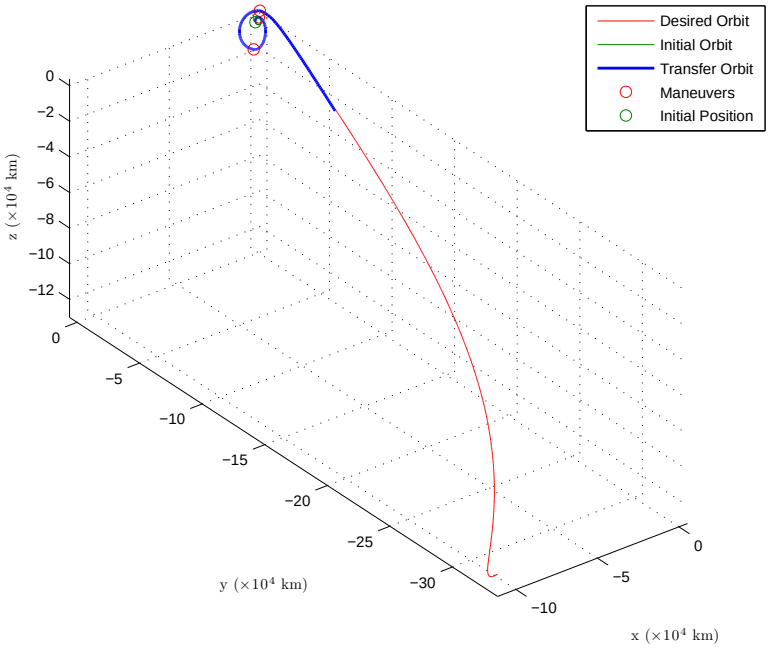


Figure 4.14: APF Lunar Return (MCI), 1.4928 km/s

the feasibility of this approach for complex trajectory design and to determine key areas of development for future research. Candidate potential functions are defined in terms of a) error between the current position and a desired intersection point on a target orbit and b) error between the current velocity and the desired velocity, and a method for calculating a desired velocity field, based on two-body analysis, is presented. These are employed in the development of a preliminary APF trajectory design algorithm for orbital transfers. This algorithm uses the time derivative of the potential to determine impulsive maneuver locations, rather than the classical APF approach of using the gradient to calculate a desired acceleration vector. The APF design method is utilized to generate sample Earth orbit and lunar return trajectories. Results indicate that this method has promise as a tool for fast calculation of startup trajectories, but the current potential function and velocity field construction limits its effectiveness in more complex dynamical systems where phasing is an important factor in the cost and structure of solution arcs. In addition, the proper selection of target conditions is a relatively unexplored area of research within the APF method that can greatly affect the quality of the initial guess. Future work will seek to address these issues through the development of potentials that incorporate more complex dynamics and timing.

Chapter 5

End-to-End Examples

In this chapter, initial guess trajectories obtained using the Artificial Potential Function (APF) method are used as startup arcs for the continuous actuation two-level targeting algorithm described in Chapter 3. Finite thrust is the assumed actuation method. These examples demonstrate how the APF method and targeting algorithm complement one another when used in conjunction. The APF method produces startup solutions that satisfy the desired entry constraints and have no substantial discontinuities, both of which improve the convergence behavior of the targeter. Conversely, the robustness and versatility of the two-level corrector can be used to compensate for weaknesses in the APF initial guess, such as high Δv or timing inaccuracies. The behavior of the targeter in response to various initial guess characteristics is also examined, in order to provide possible directions for future research.

The first step towards implementing the APF startup arcs into the two-level targeter is to select a set of patch points from the full APF trajectory. Selection of the patch points can have a significant impact on the performance of the targeting algorithm, and choosing the best set of patch states is an area of study unto itself. Thus, for this investigation, a simplified method, that

is known to produce acceptable sets of patch states, is used. Although the exact determination of patch points varies by problem, the selection procedure follows a common set of guidelines. Patch states are chosen at

- The initial state
- Each impulse state, immediately post-maneuver
- 1-2 intermediate states between maneuvers
- The end of the backwards propagation of the target state
- 2-3 intermediate states along the propagated target orbit
- The final state

This process results in 8-10 patch states for the lunar return trajectories. The full sets of patch points for the examples in this chapter may be found in Appendix B.

5.1 Lunar Return Example 1

This example explores the performance of the 2-maneuver lunar return arc, described in Chapter 4, as a startup solution. The initial states and targeted entry conditions are listed in Tables 5.1 and 5.2, respectively, and the engine parameters are the same as those given in Chapter 3 (Table 3.9). This initial arc is designed to meet all five terminal constraints with minimal discontinuities, but as discussed previously, the total Δv cost is higher than

Table 5.1: Initial Conditions

	Initial Value
Epoch	2-Aug-2018 01:53:33.8 TDT
Mass (kg)	20339.9
x (km)	-1834.7154532654
y (km)	-66.2360967947
z (km)	-73.9652632316
$\dot{x}$ (km/s)	-0.0863745295
$\dot{y}$ (km/s)	0.8139443100
$\dot{z}$ (km/s)	1.4136371915

Table 5.2: Terminal Constraints

Constraint	Value
Geodetic Altitude (km)	121.92
Longitude (deg)	-134.5456
Geocentric Latitude (deg)	-19.2041
Geocentric Azimuth (deg)	13.9960
Geocentric Flight Path Angle (deg)	-6.03

ideal. The convergence behavior of the targeting algorithm with this initial guess is examined, and a procedure for using the targeter to reduce the Δv is described and tested.

5.1.1 Finite Burn Example with Main Engine

The first sub-example assumes a high thrust engine for both maneuvers, comparable to the main engine of the MPCV. A comparison of results for both the unscaled and scaled finite burn targeting algorithm with the impulsive corrector is given in Table 5.3. The algorithm converges easily, requiring only 5 iterations for both cases (compared to 4 for the impulsive solution), but

Table 5.3: Maneuver and Convergence Data

	Impulsive	Unscaled FB	Scaled FB
TEI-1 Δv (km/s)	0.6000	0.6112	0.6106
TEI-2 Δv (km/s)	1.0989	1.0988	1.0994
Total Δv (km/s)	1.6989	1.7100	1.7100
Iterations	4	5	5

Table 5.4: Burn Data - Unscaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)
TEI-1	339.1491	3542.189
TEI-2	468.1383	4889.396
Total	807.2874	8431.585

there is an increase in cost that is almost entirely attributable to the first maneuver. From the shape of the trajectory, given in Figure 4.12 in Chapter 4, it is evident that the first maneuver occurs much closer to the Moon, raising the altitude of the vehicle until it intersects with the escape arc defined by the backward propagation from the entry conditions. It has been shown that the performance penalty for replacing an impulsive maneuver with a finite burn increases both with proximity to a gravitational body and with greater burn duration - the increase is proportional to the term $\frac{\mu \Delta t_{burn}^2}{r^3}$[42]. Thus, it is reasonable that the first maneuver, which is applied at a radius of 1837.4 km, will undergo a greater change in the transition from an impulse to a finite thrust than the second maneuver at 26761.5 km, despite the longer duration of the second burn.

Tables 5.4 and 5.5 provide a more detailed look at each converged burn

Table 5.5: Burn Data - Scaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)
TEI-1	338.8666	3539.238
TEI-2	468.4189	4892.326
Total	807.2855	8431.564

maneuver in the unscaled and scaled algorithm, respectively. The results are nearly identical, indicating that scaling does not have a significant effect in this particular problem. This suggests that the availability of feasible solutions in the vicinity of this startup arc is limited. Because the converged solution in this case is close to the initial trajectory, however, this does not present a problem in terms of algorithm performance.

5.1.2 Finite Burn Example with Auxiliary Engines

Using the same input, the lower-thrust, or auxiliary, engines are now employed for the second maneuver. The unscaled algorithm converges in 5 iterations, while the scaled version requires 7 iterations. As shown in Tables 5.6 and 5.7, the results both with and without scaling implemented are not only nearly identical to one another, but also quite similar to the high-thrust engine solution. The most noticeable difference, unsurprisingly, is the propellant mass consumption during the second maneuver, which is over 200 kg higher than in the previous example. The Δv of that burn, though, remains relatively unchanged, meaning that the Δv constraint is not necessarily limiting the fuel expenditure. It may be useful in future research to explore a direct propellant constraint rather than controlling it indirectly through the Δv.

Table 5.6: Burn Data Using Auxiliary Engines - Unscaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	339.2824	3543.581	0.6115
TEI-2	3479.0881	5111.198	1.0986
Total	3818.3705	8654.779	1.7101

Table 5.7: Burn Data Using Auxiliary Engines - Scaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	339.3463	3544.249	0.6116
TEI-2	3478.5750	5110.444	1.0984
Total	3817.9213	8654.693	1.7100

5.1.3 Cost Reduction Via Continuation

Specifications for the MPCV include a propellant mass of 8063.65 kg; using this as a benchmark, it is clear that the Δv from the previous examples is too high. The Δv of the initial guess can be lowered, however, through a continuation scheme. Using the impulsive two-level targeting algorithm, the original initial guess is converged, and that converged solution is then used as an initial guess but with a reduced Δv constraint. The entry constraints are also relaxed, allowing the entry flight path azimuth and latitude to vary. This process is repeated, using each converged solution as a new initial guess, until the desired Δv is reached. Reducing the Δv constraint by 20-30 m/s for each run ultimately yields a final solution with a Δv of 1.54 km/s. This final converged solution is then used as a startup arc in the finite burn targeting algorithm, again assuming auxiliary engines are used for the second maneuver and increasing the Δv constraint to 1.55 km/s for more flexibility. Figure

5.1 shows the converged arc of the original initial guess (black), the reduced Δv solution resulting from the continuation scheme (magenta), and the final converged solution from the finite burn targeter (blue). Patch point locations for the initial guess arcs are denoted by x's, and the final maneuver locations are shown as red circles. The converged results in Table 5.8, which are obtained after 8 iterations of the targeter, show that the total propellant mass consumed is now within the desired budget.

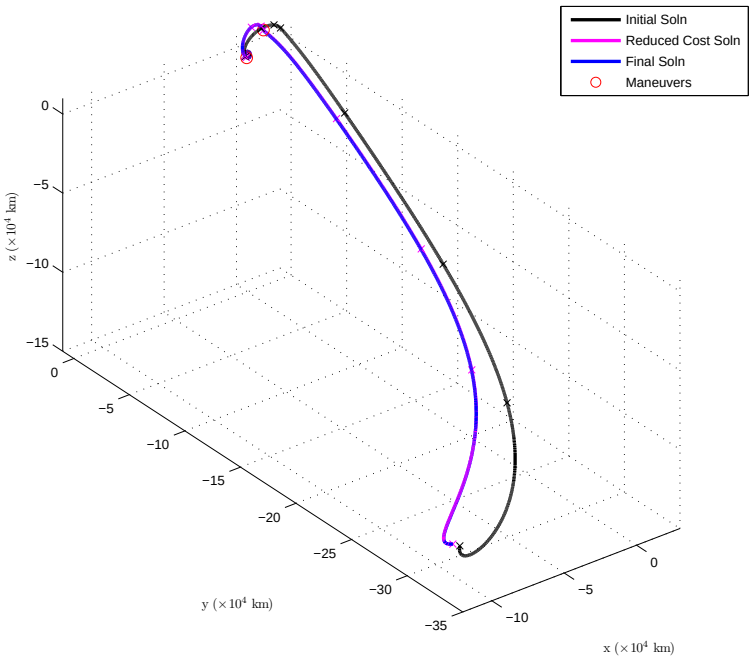


Figure 5.1: Original, Reduced Cost, and Final Converged Arcs

Table 5.8: Reduced Δv Burn Data - Unscaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	324.3184	3387.291	0.5819
TEI-2	3156.0145	4636.564	0.9674
Total	3480.3329	8023.855	1.5493

Table 5.9: Initial Conditions

	Initial Value
Epoch	2-Aug-2018 19:51:46.0 TDT
Mass (kg)	20339.9 (total fuel = 8063.65 kg)
x (km)	-1834.7154532654
y (km)	-66.2360967947
z (km)	-73.9652632316
$\dot{x}$ (km/s)	-0.0863745295
$\dot{y}$ (km/s)	0.8139443100
$\dot{z}$ (km/s)	1.4136371915

5.2 Lunar Return Example 2

In this second example, the 3-maneuver lunar return trajectory from Chapter 4 is employed as an initial guess in the finite burn targeter. The initial and target terminal conditions are given in Tables 5.9 and 5.10. Note that now, a strict total fuel limit of 8063.65 kg is enforced in place of the entry flight-path azimuth. As mentioned before, the entry epoch and flight-path angle have been modified slightly from the previous example, but the flight-path angle constraint is still imposed. Like the first case, this example evaluates the targeting algorithm performance with both high-thrust and lower-thrust engines, and with and without variable scaling implemented. The initial time shift of the startup arc, and a means of mitigating that, is also considered.

Table 5.10: Terminal Constraints

Constraint	Value
Geodetic Altitude (km)	121.92
Longitude (deg)	-134.5456
Geocentric Latitude (deg)	-19.2041
Geocentric Flight Path Angle (deg)	-5.86

5.2.1 Finite Burn Example with Main Engine

The first case examined again uses high-thrust engines for all of the burn maneuvers. Table 5.11 shows a comparison of the impulsive, unscaled finite burn, and scaled finite burn results. Unlike the previous example, this startup arc displays a noticeable sensitivity to the implementation of variable scaling, indicating a greater availability of nearby solutions. The scaled version of the algorithm demonstrates both a faster convergence (3 iterations compared to 6) and a lower total Δv cost. From the individual maneuver Δv's in Table 5.11, the scaled algorithm converges to a higher Δv for the initial maneuver, but this ultimately results in a significantly lower Δv at the third maneuver. Recall the rocket equation (Equation 3.39, in Chapter 3). For a constant thrust engine, the only dependent variables affecting the Δv of a burn are the burn duration, Δt_{burn}, and the initial mass m_k. Employing a more substantial burn at the first maneuver reduces the total mass of the vehicle, which, from Equation 3.40, also reduces the burn durations of the later maneuvers. Although increasing the initial burn may in fact result in a more favorable trajectory geometry, the decreased burn duration certainly contributes to the lower overall Δv of the scaled algorithm solution.

Table 5.11: Maneuver and Convergence Data

	Impulsive	Unscaled FB	Scaled FB
TEI-1 Δv (km/s)	0.5438	0.5490	0.5568
TEI-2 Δv (km/s)	0.2912	0.2930	0.2941
TEI-3 Δv (km/s)	0.6651	0.6581	0.6223
Total Δv (km/s)	1.5001	1.5001	1.4732
Iterations	4	6	3

Table 5.12: Burn Data - Unscaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)
TEI-1	307.5251	3211.897
TEI-2	143.7250	1501.113
TEI-3	278.5624	2909.400
Total	729.8125	7622.410

More insight can be gained by examining the details of each burn maneuver, listed in Tables 5.12 and 5.13. Because of the high engine thrust, and thus the high propellant mass flow rate (assuming a constant I_{sp}), the reduction in burn duration at the third maneuver results in significant savings in fuel consumption. By making the first burn just 4 seconds longer, and with essentially no change to the second burn, the final burn duration decreases by 14 seconds. This results in a net reduction of nearly 30 m/s of Δv and over 100 kg of propellant mass consumption. Based on these results, it may be worthwhile to explore potential function constructions that encourage larger initial maneuvers in the initial guess arc, despite the possibility of increased Δv penalties in the transition from impulsive to finite burn maneuvers.

Table 5.13: Burn Data - Scaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)
TEI-1	311.5061	3253.475
TEI-2	143.9096	1503.041
TEI-3	264.1220	2758.580
Total	719.5377	7515.096

5.2.2 Finite Burn Example with Auxiliary Engines

Here, for the same initial arc, the lower-thrust auxiliary engines are substituted for the final two burn maneuvers. Burn data for the unscaled and scaled versions of the finite burn targeting algorithm, both of which converge in 4 iterations, is given in Table 5.14 and Table 5.15. For this case, although the scaled algorithm still performs better overall, the difference is much less pronounced. Compared to the impulsive results from the previous section, both versions of the algorithm converge with a higher Δv at the first maneuver, possibly to offset the much longer burn durations required at the subsequent maneuvers. It is interesting to note that, despite the fact that both the scaled and unscaled algorithm converge to lower total Δv values than in the previous case, the fuel consumption is actually higher in both cases than either of the main engine results. The lower mass flow rate is not sufficient to compensate for the substantially longer burn durations. As with the two-maneuver example, this suggests that future alterations to the finite burn algorithm should consider constraining the propellant consumption directly, rather than through the Δv constraint.

Table 5.14: Burn Data Using Auxiliary Engines - Unscaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	310.6551	3244.588	0.5551
TEI-2	1087.6472	1597.884	0.2971
TEI-3	1935.9439	2844.134	0.6139
Total	3334.2462	7686.606	1.4661

Table 5.15: Burn Data Using Auxiliary Engines - Scaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	311.8613	3257.186	0.5574
TEI-2	1081.6031	1589.005	0.2956
TEI-3	1908.7916	2804.244	0.6045
Total	3302.2560	7650.435	1.4575

One thing to note in both this and the previous case is that there is less discrepancy between the impulsive solution and the finite burn solutions than there was for the two-maneuver startup arc. With three smaller maneuvers, the burn durations at each maneuver are lower, and thus the Δv increase due to the finite burn is lessened. This is something to consider for future iterations of the artificial potential function design algorithm.

5.2.3 Time-Shifted Initial Guess

One drawback to the APF method, as mentioned before, is that the initial time is not constrained to match the original input initial time. If the targeting process is done prior to launch, this is not a significant issue, but for the approach to be useful for onboard applications it must be addressed either in the initial guess procedure or by the targeting algorithm. As demonstrated

in Chapter 3, the finite burn two-level corrector is capable of converging a time-shifted initial guess. One approach, then, would be to shift all of the patch points for the 3-maneuver initial guess such that the start time of the trajectory matches the original t_0. While valid, this is not necessarily the best method for recovering the correct initial time. If these time-shifted patch points are fed directly into the finite burn targeter, assuming the auxiliary engines are used for the last two maneuvers, it requires 18 iterations for the algorithm to converge even if only the altitude and flight path angle are constrained. In the impulsive two-level corrector, 11 iterations are required to converge the time-shifted trajectory with only altitude and flight-path angle targeted. If continuation is used to target the latitude and longitude as well, another 4 iterations give a trajectory with the correct entry longitude, and an additional 11 iterations are required to find a solution with the entry latitude constrained as well. From the initial and final trajectories (i.e., before and after continuation) shown in Figure 5.2, it is evident that the targeting algorithm must substantially change the startup solution in order to meet the desired entry constraints.

Due to the nature of the APF process, however, it is not actually necessary to shift all of the patch points, nor even to target the entry constraints directly. Because a segment of the trajectory comes from propagating the entry state backwards, it is actually only necessary to target the final state of that propagation - in other words, the interface point between the back propagated trajectory and the APF-generated trajectory. If only the APF segment

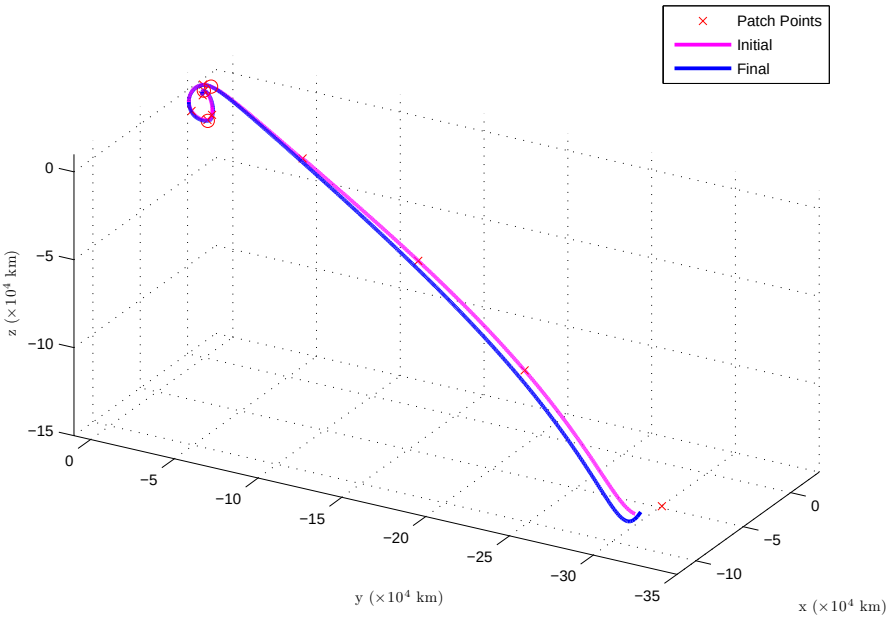


Figure 5.2: Time-Shifted Lunar Return (MCI), 1.5000 km/s

of the trajectory is time-shifted, and the target constraints are defined as the interface position, velocity, and time, then the impulsive corrector converges in 14 iterations, rather than the 26 total iterations required by the previous method. This approach, though, can also be improved upon. Since the entry speed and flight path azimuth are allowed to vary in this example, it is advantageous to use the full trajectory, rather than just the APF arc, as a startup solution, and to only target the four entry constraints as before. This allows the targeter more flexibility in finding a feasible solution. However, instead of shifting the times for all of the patch points, only the lunar arc (APF segment) patch points are altered. The rest are left unchanged, since they are known to match the desired entry conditions. Under these circumstances, the impulsive algorithm converges in 11 iterations; the initial and final arcs are shown in Figure 5.3. Clearly, this approach requires a much less significant change to the startup arc than the first method (Figure 5.2), and thus it converges considerably more quickly.

The final resulting trajectory, with all four entry constraints met, is then used in the finite burn targeter as an initial guess. The results in Tables 5.16 and 5.17 are for the auxiliary engine case. Unscaled, the algorithm converges in 3 iterations, while the scaled algorithm requires 7 iterations.

5.3 Summary

This chapter presents a synthesis of the two algorithms described in this work, a continuous actuation targeting algorithm and an artificial potential

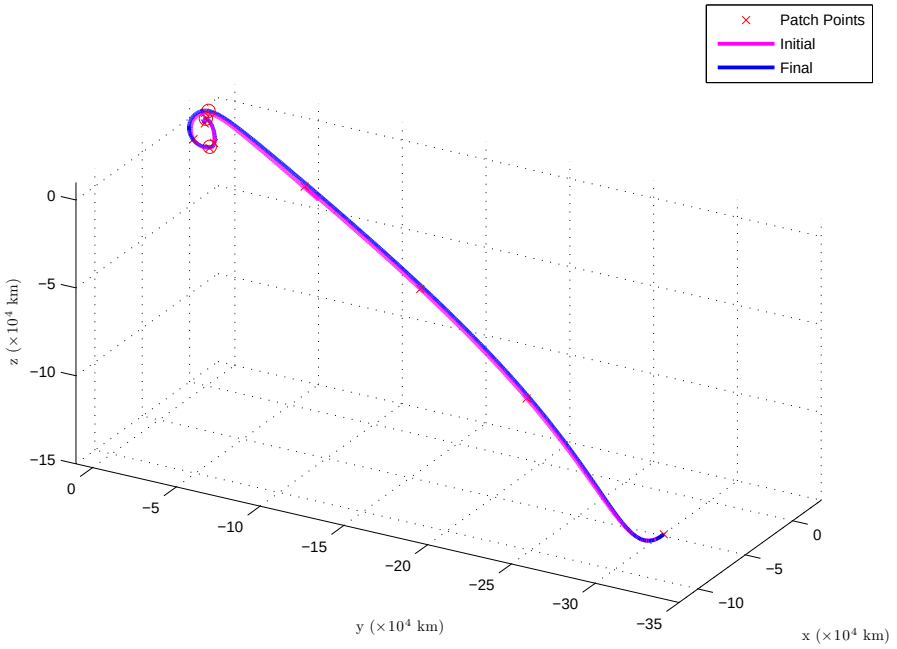


Figure 5.3: Partially Time-Shifted Lunar Return (MCI), 1.5000 km/s

Table 5.16: Burn Data Using Auxiliary Engines - Unscaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	309.7365	3234.993	0.5533
TEI-2	1207.3625	1773.760	0.3315
TEI-3	1918.9526	2819.171	0.6152
Total	3436.0516	7827.924	1.5000

Table 5.17: Burn Data Using Auxiliary Engines - Scaled Algorithm

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	308.8013	3225.225	0.5515
TEI-2	1159.4937	1703.435	0.3174
TEI-3	1973.8934	2899.886	0.6311
Total	3442.1884	7828.546	1.5000

function trajectory design algorithm. Although each methodology can be used independently of the other for various applications, the primary goal of this study is the development of a self-contained targeting tool capable of both autonomously generating a startup solution and targeting the specified path constraints. The examples shown here demonstrate the end-to-end automated guidance procedure, beginning with the identification of a startup arc via the APF algorithm and followed by the targeting process using a modified two-level corrector. The initial guess arcs are designed to meet any required terminal constraints and to have minimal discontinuities, both of which improve the convergence behavior of the targeter. Likewise, the targeter is shown to be capable of addressing any undesirable traits of the startup arc, such as high Δv or start time inaccuracies.

In addition to demonstrating the combined performance of the APF method and the targeter, the results presented here also provide insight into possible future research efforts that may enhance the effectiveness of this tool. For the targeter, these include the development of a fuel consumption constraint and more efficient methods of dealing with shifts in the initial time of the trajectory. Of interest for the APF algorithm would be an exploration of the impact of the number and location of maneuvers, and their relative magnitudes, on the transition from an impulsive solution to a finite burn solution.

Chapter 6

Conclusions

This work presents the development of computationally efficient targeting and preliminary trajectory identification algorithms that may be used in conjunction to serve as an onboard, fully autonomous guidance tool for spacecraft applications. The methodologies described address the two primary requirements of autonomy as defined in this study: (1) the automatic determination of an appropriate startup solution, and (2) the ability to target, using only the available actuation method and budget, any specified terminal and path constraints. The efficacy of each of the two algorithms is demonstrated through several examples, both separately and conjointly as an end-to-end procedure.

The investigation focuses first on targeting the path and terminal constraints, specifically for continuous or piecewise continuous actuation methods. The approach is based on the impulsive two-level targeting algorithm, but includes several modifications to allow for the incorporation of continuous actuation. This new algorithm preserves the robustness and efficiency of the classical implementation while greatly expanding the spectrum of potential applications.

The key distinction enabling the incorporation of continuous actuation methods is the augmentation of the state vector to include the control parameters, so that the differential equation governing the controlled system, including any actuator dynamics, may be written as a function of the state and time only. The linearized dynamical model then takes the form of a zero-input system, which is consistent with the classical implementation of the two-level corrector. Another modification is the introduction of sub-arcs within each pre-specified trajectory segment. The initial and final times of each sub-arc represent times of instantaneous changes in one or more control parameters, similar to switching times in optimal control theory, and are used as control variables in the two-level targeter framework.

Because the continuous actuation targeting algorithm follows the same structure as the classical two-level corrections process, it retains many of the inherent advantages of that method. It requires no knowledge of a nominal path, only an initial set of patch states to represent the startup solution. The addition of constraints, whether applied at a particular point or over the entire trajectory, is straightforward, and the use of a linearized model gives it significant computational simplicity and efficiency. Furthermore, the inclusion of sub-arcs with variable terminal times in the new algorithm allows for an increased number of control variables in the Level I procedure. This in turn increases the number of constraints that may be simultaneously targeted during that process.

A final advantage is that this method is not model-dependent, and

may thus be implemented for a number of different systems. This is validated through several sets of examples in Chapter 3, building in actuator complexity from a constant torque input to reaction wheels to finite thrust. The applications primarily showcase the Level I process, as that is where the majority of the modifications take place, but several examples implementing the full two-level procedure are provided as well.

The second part of the investigation considers the automatic determination of startup solutions. Artificial potential function (APF) methods, which are prevalent in path planning for other applications, are examined as a tool for preliminary trajectory design. The specific methodology utilized is a velocity error-based potential function that relies on the computation of a desired velocity field to define the motion of the vehicle. The resulting trajectories closely match the correct terminal constraints and display minimal discontinuities, but may have timing or actuation budget issues that must be addressed by the subsequent targeting algorithm.

Potential function methods have been used for many years to perform path planning for ground-based robots, attitude maneuvers, and spacecraft formation flight, among other applications. Because these methods typically assume the availability of a continuously applied, unconstrained control input, they are at first glance not well-suited for general trajectory design. However, the implementation of a desired velocity field, computed using two-body approximations, and velocity error-based potential mitigates these issues significantly. Instead of using the potential field to determine the control acceler-

ation, this algorithm seeks the minimum of the potential with respect to time in order to identify suitable impulsive maneuver locations. Because this value of the potential is based on the difference between the current velocity and the desired velocity at that point, this serves to drive down the overall Δv cost of the trajectory.

A notable advantage of the APF approach, in addition to the speed and computational simplicity inherent in the methodology, is its versatility. Trajectory characteristics in addition to the cost, such as a low transfer time or specified minimum or maximum periapse altitude, may be incorporated individually or simultaneously into the construction of the potential to produce desirable behavior in the startup arc. An example in Chapter 4 demonstrates how this method may be used to generate a simple coplanar transfer arc that is both low-cost and short in duration. Other advantages include minimal discontinuities and close matching of any imposed terminal constraints, as shown particularly by two precision entry lunar return examples.

Finally, several examples are presented to show how the continuous actuation targeting algorithm and APF startup arc identification method work together as a self-contained autonomous targeting tool. These examples first demonstrate the good convergence behavior of the targeting algorithm when seeded with APF-produced initial solutions; the targeter identifies a feasible solution in relatively few iterations while meeting every specified constraint to within the desired tolerance. Further examples show how any drawbacks to the potential function solution, such as a high cost or initial timing error, may

be addressed by the targeting algorithm without significantly increasing the computational requirement.

6.1 Suggested Future Work

There are several research areas in which future work may enhance the methodologies developed in this investigation. Some of these are listed below.

- Continuous Actuation Targeting
  - Several formulations currently exist within the two-level targeter framework for constraints that are imposed at a single patch state. These are primarily implemented as terminal constraints, but are not limited to be so. However, only one constraint, the total cost, can be applied over the entire trajectory. The ability to incorporate similar constraints, that are not necessarily associated with a particular patch point, would broaden the range of possible applications for this algorithm and perhaps improve the solution process for certain scenarios in the current applications. Specific examples include a propellant consumption constraint or an obstacle-avoidance constraint.
  - For good convergence behavior in the targeting algorithm, it is essential for the State Relationship Matrix (SRM) to be well-conditioned. Investigating the threshold of SRM condition number for which convergence becomes inhibited, and furthermore the effects of variable

scaling on the SRM condition number, could give valuable insight that may improve the performance of the targeter.

- Preliminary Trajectory Identification
  - Because it is a linear correction process, the targeting algorithm works best when the desired perturbations to the state are in or near the linear range. For highly nonlinear systems, such as the attitude slew example in Chapter 3, the linear approximation can break down fairly quickly. Thus, the selection of appropriate patch states is an important aspect of determining a startup solution. In this study, the patch states are selected heuristically. An investigation into the relationship between the system dynamics, patch point locations, and targeter convergence behavior could provide a more reliable and generalized selection method.
  - In its current form, the APF design algorithm relies on the knowledge of a full terminal state vector in order to determine the target orbit for velocity field computation. The ability to autonomously determine an appropriate terminal state from just one or two constraints would enhance the generality of the approach.
  - The velocity field computations developed in this study utilize two-body approximations. A more complex formulation, perhaps incorporating third body, spherical harmonics, or phasing effects, may provide better results for identifying cislunar or interplanetary tra-

jectories. Also, an analysis of the transition from impulsive to finite burn maneuvers could provide additional considerations for potential function constructions, leading to startup arcs that convert more readily to the finite burn case.

Appendices

Appendix A

Chapter 3 Initial Patch States

A.1 Precision Entry Examples

Table A.1: Patch Points for Precision Entry (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	0.0000	-1236.7971	1268.1142	468.3832	0.0329	0.5893	-1.5281
2	0.1005	-121.3092	815.2075	-1634.4365	1.1019	-1.0515	-0.6104
3	0.1911	2306.1633	-7378.9082	3627.3188	0.0948	-0.4817	0.7548
4	0.7049	1835.6098	-11419.4236	22195.6003	-0.0936	0.0351	0.1987
5	1.3559	-3476.1712	-4987.7369	22618.7012	-0.0805	0.1797	-0.1975
6	1.3778	-3626.6133	-4643.3008	22227.9938	-0.0781	0.1834	-0.2145
7	1.5239	-1297.0258	-1021.2100	17645.2059	0.1940	0.2965	-0.4452
8	1.7621	2213.8423	3897.3755	2497.7762	-0.0115	-0.1042	-1.2933
9	1.7760	2131.5697	3652.8726	886.4852	-0.1355	-0.3196	-1.3861
10	2.9295	-43275.5730	-74480.9870	-67695.4480	-0.4413	-0.7158	-0.5660
11	3.9547	-85232.2773	-132990.7270	-113066.7120	-0.5333	-0.5964	-0.4461
12	5.7457	-271969.5636	-219843.5338	-113476.0288	-1.5644	-8.7906	6.6781

A.2 Lunar Cycle Examples

Table A.2: Patch Points for Lunar Cycle Example, Day 1 (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	0.0800	1811.7587	195.0894	-235.6049	0.2603	-0.6192	1.4890
2	0.0939	1131.3087	-515.7179	1352.8625	-1.2847	-0.4509	0.9025
3	0.3186	-14025.7177	1927.5216	-6280.2732	-0.3177	0.1572	-0.4112
4	0.9791	-17191.7336	7312.5906	-19842.2331	0.1021	0.0379	-0.0924
5	0.9930	-17066.2790	7356.7247	-19949.6089	0.1069	0.0356	-0.0866
6	1.2046	-9351.0629	1556.4313	-14038.6710	0.4850	-0.3357	0.4116
7	1.4023	976.2354	-3258.8246	-2346.6280	0.6946	0.1290	1.3460
8	1.4161	1731.7989	-2909.4684	-624.9496	0.5363	0.4787	1.5064
9	1.8745	-7631.4693	35017.6465	30878.2008	-0.2610	0.8707	0.6288
10	4.3982	-68176.3260	211519.7031	63882.4925	-0.0228	0.8257	0.3968
11	5.6600	-6492.4424	331187.5816	178399.6460	-3.4414	-1.6887	10.5794

Table A.3: Patch Points for Lunar Cycle Example, Day 3 (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	0.0843	1808.3660	-42.3341	322.5813	-0.2783	-0.6419	1.4761
2	0.0982	599.1122	-652.7208	1609.6766	-1.5422	-0.2770	0.4617
3	0.3230	-11128.2055	3698.8282	-10101.5831	-0.1561	0.1947	-0.4808
4	0.9834	-9058.5751	9244.4824	-23530.4062	0.1395	0.0193	-0.0459
5	0.9973	-8889.6073	9265.8232	-23581.1814	0.1421	0.0163	-0.0387
6	1.2098	-2797.5015	3375.4149	-15598.5634	0.3588	-0.3526	0.5462
7	1.4084	2404.2421	-2174.2885	66.8395	-0.3267	0.2024	1.6804
8	1.4223	1762.4891	-1701.3856	1999.7540	-0.7280	0.5758	1.4787
9	1.8784	-31505.2873	26332.2563	28434.0208	-0.7656	0.6567	0.5435
10	4.3995	-173760.0319	158396.9655	36482.3868	-0.4494	0.6939	0.3268
11	5.6600	-181748.3946	277178.0908	154541.5980	-3.2206	-4.3371	9.7804

Table A.4: Patch Points for Lunar Cycle Example, Day 6 (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	0.0861	1748.1314	-141.2503	547.8324	-0.4968	-0.6307	1.4226
2	0.1000	354.8423	-689.4176	1665.7671	-1.6008	-0.1946	0.2605
3	0.3248	-9533.4536	4333.1892	-11380.9432	-0.0816	0.2049	-0.4937
4	0.9852	-5183.2054	9853.0910	-24131.4075	0.1532	0.0123	-0.0189
5	0.9991	-4998.3517	9865.8938	-24149.4498	0.1549	0.0090	-0.0112
6	1.2117	713.1056	5701.0557	-14614.7564	0.3160	-0.2733	0.6350
7	1.4104	2152.0911	-952.3354	2005.8320	-1.2197	-0.3889	1.2195
8	1.4243	504.8435	-1299.1310	3200.0587	-1.4641	-0.1912	0.7644
9	1.8802	-47750.6101	-1409.9080	12333.5821	-1.0937	0.0215	0.1620
10	4.4001	-245843.4912	-42627.8549	38221.0258	-0.8456	0.1536	0.1856
11	5.6600	-349455.9999	77757.5019	52025.1962	-1.5528	4.0371	-10.2007

Table A.5: Patch Points for Lunar Cycle Example, Day 10 (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	0.0926	1264.9187	-464.7081	1249.0292	-1.1822	-0.4929	1.0138
2	0.1065	-553.5780	-709.9509	1601.7088	-1.5556	0.1237	-0.4828
3	0.3313	-2530.9056	5861.9835	-14085.8017	0.1842	0.2073	-0.4637
4	0.9917	8028.4844	10212.7825	-23136.9288	0.1501	-0.0181	0.0532
5	1.0056	8207.1566	10189.0496	-23068.6637	0.1477	-0.0215	0.0606
6	1.2323	7342.8158	8139.5217	-11552.4736	-0.0959	-0.1666	0.6902
7	1.4451	31.4721	-330.8582	3115.5105	-1.2740	-1.1851	0.0165
8	1.4590	-1450.9126	-1672.7051	2814.0091	-1.1643	-1.0260	-0.4780
9	3.1418	-56026.8814	-127020.3316	-34394.5031	-0.4214	-0.8230	-0.2566
10	4.4009	-112061.4976	-204518.5425	-59686.8804	-0.6789	-0.5739	-0.2113
11	5.6600	-275896.0981	-231377.9901	-116889.2677	-1.0276	-0.2496	-11.2250

Table A.6: Patch Points for Lunar Cycle Example, Day 13 (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	0.1101	-1008.9920	-645.1486	1393.4239	-1.3629	0.2908	-0.8523
2	0.1240	-1829.5781	-25.0520	-166.6472	0.1275	0.6427	-1.4964
3	0.3488	14295.4678	2950.5652	-5172.8982	0.5397	-0.0011	0.0702
4	1.0092	27686.4726	610.6210	2390.6729	0.0255	-0.0530	0.1434
5	1.0231	27712.7559	547.0105	2562.4190	0.0183	-0.0530	0.1428
6	1.2501	17675.3814	3587.5500	3941.6792	-0.6108	0.1426	0.0529
7	1.4633	2749.0400	4655.2567	3513.0918	-1.1333	-0.2042	-0.2348
8	1.4772	1354.5350	4340.4663	3179.1459	-1.1886	-0.3284	-0.3271
9	1.9001	-798.5047	-31101.8060	-21962.1362	0.1394	-0.8250	-0.5670
10	3.1534	33319.8743	-131620.3865	-37122.6207	0.1975	-0.8889	-0.3178
11	4.4067	39648.2384	-226304.1089	-74508.5354	-0.1446	-0.8816	-0.4026
12	5.6600	-49527.5585	-345067.8043	-184830.5590	2.9658	0.7985	-10.8586

Table A.7: Patch Points for Lunar Cycle Example, Day 16 (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	0.1251	-1811.5619	34.6015	-304.7882	0.2611	0.6423	-1.4791
2	0.1389	-617.5457	649.3123	-1604.0504	1.5365	0.2832	-0.4769
3	0.3637	11241.1289	-3645.1334	9993.3322	0.1614	-0.1936	0.4793
4	1.0241	9277.4425	-9157.9284	23429.3105	-0.1404	-0.0189	0.0466
5	1.0380	9107.3016	-9178.7702	23480.8095	-0.1432	-0.0159	0.0393
6	1.2563	2910.7369	-3226.4340	15660.7124	-0.3572	0.3474	-0.5287
7	1.4607	-2399.1281	2229.3488	-257.6442	0.3423	-0.2594	-1.6510
8	1.4746	-1753.2367	1697.0191	-2141.7328	0.7176	-0.6144	-1.4330
9	1.9172	30557.8035	-26702.9250	-28822.8324	0.7709	-0.6892	-0.5735
10	3.1648	109867.2940	-98028.9783	-14839.0406	0.7427	-0.6315	-0.1943
11	4.4124	178875.1209	-173032.7415	-43878.0330	0.5081	-0.8075	-0.3910
12	5.6600	202052.8349	-306812.7946	-171394.3043	2.8040	4.7618	-9.6949

Table A.8: Patch Points for Lunar Cycle Example, Day 19 (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	0.1315	-1458.6933	371.4254	-1053.6583	0.9901	0.5518	-1.1762
2	0.1453	270.8822	722.8898	-1667.3461	1.6137	-0.0227	0.2523
3	0.3701	4896.0745	-5518.8358	13593.6064	-0.1024	-0.2120	0.4869
4	1.0305	-3861.9829	-10396.1004	24210.9196	-0.1519	0.0070	-0.0277
5	1.0444	-4043.5668	-10385.6517	24173.0226	-0.1507	0.0104	-0.0354
6	1.2483	-5790.2186	-5342.4929	13529.3366	-0.0641	0.3302	-0.7104
7	1.4383	-1808.0871	1245.7800	-2202.9813	1.4711	0.2359	-0.9621
8	1.4522	95.5490	1378.6353	-3067.9724	1.6381	-0.0088	-0.4729
9	1.9296	49289.9464	-6908.5525	509.3774	1.0412	-0.2032	0.1232
10	3.1731	146239.2488	8197.9877	-44201.0597	0.9807	-0.0034	-0.1890
11	4.4165	249252.0901	-6919.0600	-67598.9686	0.9514	-0.3469	-0.2620
12	5.6600	374034.9476	-143494.1636	-87995.2489	1.9669	-4.0490	10.1518

Table A.9: Patch Points for Lunar Cycle Example, Day 22 (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	0.1324	-1372.7642	416.1739	-1148.1596	1.0829	0.5261	-1.1041
2	0.1463	403.7304	719.1693	-1641.8973	1.5916	-0.0699	0.3607
3	0.3711	3807.9695	-5699.6526	13865.8204	-0.1409	-0.2108	0.4776
4	1.0315	-5804.0102	-10410.4652	23838.0931	-0.1513	0.0104	-0.0375
5	1.0454	-5984.4827	-10395.8924	23788.6012	-0.1495	0.0138	-0.0450
6	1.2540	-6486.6708	-7362.3226	12369.9484	0.0134	0.2225	-0.7339
7	1.4487	-1343.7399	47.4532	-2790.3132	1.3866	0.9574	-0.5431
8	1.4625	411.5817	1147.8427	-3098.3589	1.4817	0.8469	0.0188
9	1.9282	42774.2210	20266.0156	15307.5534	0.9216	0.3952	0.4504
10	4.4161	215687.5004	114836.2119	2609.5821	1.0142	0.2054	0.0438
11	5.6600	400920.3268	75545.6746	29789.7197	2.5928	-1.0520	10.9082

Table A.10: Patch Points for Lunar Cycle Example, Day 25 (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	0.1365	-930.3655	578.3179	-1475.1075	1.4064	0.3864	-0.7356
2	0.1504	936.0004	659.8702	-1436.8493	1.4034	-0.2636	0.7932
3	0.3752	-992.6047	-6084.1095	14185.9253	-0.2949	-0.1904	0.4114
4	1.0356	-13691.6517	-9564.2791	20920.7058	-0.1387	0.0272	-0.0764
5	1.0495	-13855.6951	-9529.8543	20825.0577	-0.1347	0.0302	-0.0830
6	1.2606	-9545.5014	-8153.0693	10076.5928	0.3025	0.1300	-0.6710
7	1.4579	-227.9130	-1337.3101	-2929.2129	1.1238	1.2728	-0.3223
8	1.4718	1094.9830	279.8446	-2978.7441	1.0397	1.3733	0.2450
9	1.9312	22181.3547	37064.2744	24920.0390	0.4456	0.7988	0.6692
10	4.4171	110471.2662	200681.6722	54605.5021	0.6789	0.6752	0.3105
11	5.6600	270356.3564	262953.8859	134978.4749	-0.3476	0.5289	11.2915

Table A.11: Patch Points for Lunar Cycle Example, Day 28 (MCI)

k	**Time (d)**	x **(km)**	y **(km)**	z **(km)**	v_x **(km/s)**	v_y **(km/s)**	v_z **(km/s)**
1	0.0785	1766.1394	273.7999	-426.4110	0.4436	-0.5952	1.4551
2	0.0924	1289.7995	-454.0580	1227.3176	-1.1607	-0.5006	1.0346
3	0.3171	-14683.0470	1252.2944	-4770.0449	-0.3663	0.1398	-0.3761
4	0.9776	-19575.8699	6424.7692	-18002.0623	0.0870	0.0430	-0.1052
5	0.9915	-19468.2004	6475.1207	-18125.1993	0.0924	0.0409	-0.1000
6	1.2031	-11110.1643	985.3447	-13030.7577	0.5286	-0.3146	0.3576
7	1.4009	418.9199	-3437.3750	-2815.4530	0.8540	0.1052	1.1791
8	1.4148	1405.0032	-3146.6234	-1289.1986	0.7665	0.4038	1.3594
9	1.8730	-4139.8111	35103.6917	29366.2119	-0.1933	0.8735	0.6036
10	4.3977	-45938.1764	218099.6459	67455.5807	0.0647	0.8437	0.4108
11	5.6600	27805.4616	337384.0539	182113.0290	-3.3113	-1.3542	10.6807

Appendix B

Artificial Potential Function Patch States

B.1 Two-Maneuver Lunar Return

Table B.1: Baseline Two-Maneuver Lunar Return (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	-0.6407	-1834.7155	-66.2361	-73.9653	-0.0864	0.8139	1.4136
2	-0.5617	-1771.7241	-260.0766	-411.5752	-0.5913	1.0749	1.8663
3	-0.2841	13690.8650	5490.0003	9175.3697	0.4632	0.0347	0.0492
4	0.0143	22298.5041	5252.3848	8521.1316	0.2265	-0.0396	-0.0773
5	0.2906	25781.9572	3922.9684	6006.5348	0.1789	-1.0551	-0.5751
6	0.9005	28512.8781	-49541.6365	-24013.3720	0.0079	-0.9762	-0.5555
7	2.0535	25892.4395	-141842.7493	-76573.9578	-0.0771	-0.8861	-0.4990
8	3.1552	7175.5662	-223835.7868	-120152.2007	-0.3799	-0.8521	-0.4025
9	4.3167	-111210.3370	-335970.3795	-117744.8222	-0.6583	-3.3506	10.2435

Table B.2: Δv-Reduced Two-Maneuver Lunar Return (MCI)

k	Time (d)	x (km)	y (km)	z (km)	v_x (km/s)	v_y (km/s)	v_z (km/s)
1	-0.6411	-2155.2202	379.2277	686.1397	-0.0865	0.8140	1.4137
2	-0.6311	-1970.0486	1009.8300	1778.0993	0.4648	0.8678	1.4581
3	-0.2965	14686.3618	6575.0512	10673.2486	0.3879	0.0242	0.0272
4	0.0077	21981.1374	6086.1929	9534.6312	0.1795	-0.0508	-0.0965
5	0.2949	24368.9564	4334.1401	6327.0721	0.1866	-1.0712	-0.5781
6	0.8978	27743.6439	-49200.6569	-23646.0637	0.0117	-0.9762	-0.5564
7	2.0516	25467.6152	-141541.5992	-76322.3745	-0.0739	-0.8860	-0.4998
8	3.1545	7015.2128	-223634.7603	-120035.7895	-0.3772	-0.8525	-0.4035
9	4.3168	-111194.5839	-335979.1837	-117747.0386	-0.6583	-3.3505	10.2435

B.2 Three-Maneuver Lunar Return

Table B.3: Baseline Three-Maneuver Lunar Return (MCI)

k	**Time (d)**	x **(km)**	y **(km)**	z **(km)**	v_x **(km/s)**	v_y **(km/s)**	v_z **(km/s)**
1	0.1081	-1834.7155	-66.2361	-73.9653	-0.0864	0.8139	1.4136
2	0.1365	968.3459	789.2647	1347.3274	1.8482	-0.5780	-0.9898
3	0.3203	-2092.5212	-5760.0451	-9839.0314	-0.4016	-0.1403	-0.2391
4	0.5131	-7553.0559	-6139.4101	-10468.1473	-0.2622	0.3358	-0.0077
5	0.7599	-9406.4376	2369.0033	-6504.1386	0.1440	0.4037	0.3888
6	1.0106	2652.7881	2265.2132	3771.4844	0.7332	-1.5551	-0.1572
7	1.6253	4656.4792	-55991.4438	-26215.9384	-0.0238	-0.9673	-0.5515
8	2.5070	-327.1903	-127861.9813	-66767.0574	-0.1267	-0.9356	-0.5150
9	3.4021	-20274.8296	-202602.9926	-104884.3848	-0.4506	-1.0349	-0.4639
10	4.3174	-111139.8800	-336009.7994	-117754.6122	1.4680	-8.0218	7.5642

Table B.4: Time-Shifted Three-Maneuver Lunar Return (MCI)

k	**Time (d)**	x **(km)**	y **(km)**	z **(km)**	v_x **(km/s)**	v_y **(km/s)**	v_z **(km/s)**
1	0.0000	-1834.7155	-66.2361	-73.9653	-0.0863	0.8140	1.4137
2	0.0278	898.2565	809.2448	1383.5437	1.8992	-0.5834	-0.8687
3	0.2143	-1669.6178	-5819.3418	-9707.8677	-0.3909	-0.1274	-0.2302
4	0.4019	-6902.1306	-5890.6025	-10079.7995	-0.3249	0.3532	-0.0217
5	0.6468	-9844.3628	2924.3542	-6200.9973	0.0966	0.4125	0.3816
6	0.9670	4311.5402	1156.7751	4496.9309	0.5580	-1.4852	-0.3420
7	1.6369	6490.8609	-61002.6381	-28486.6726	-0.0248	-0.9563	-0.5483
8	2.5072	1407.3021	-131264.1164	-68349.2906	-0.1298	-0.9261	-0.5128
9	3.4021	-18925.9320	-205084.4666	-106190.8956	-0.4583	-1.0200	-0.4587
10	4.3163	-111245.4984	-335950.6487	-117739.9633	1.4680	-8.0220	7.5641

Bibliography

[1] Jasim Ahmed, Vincent T. Coppola, and Dennis S. Bernstein. Adaptive asymptotic tracking of spacecraft attitude motion with inertia matrix identification. *Journal of Guidance, Control, and Dynamics*, 21(5):684–691, Sep-Oct 1998.

[2] A.H. Bajodah and D.H. Hodges. Linear parameterization of nonlinear spacecraft control by pseudoinversion of the control coeffcient. In *AIAA Guidance, Navigation, and Control Conference*, San Francisco, CA, Aug. 2005. AIAA 2005-6181.

[3] J. M. Baker. Orbit transfer and rendezvous maneuvers between inclined circular orbits. *Journal of Spacecraft and Rockets*, 3(8):1216–1220, 1966.

[4] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*. AIAA, Reston, VA, 1999.

[5] R. A. Broucke and A. F. B. A. Prado. Optimal N-Impulse Transfer Between Coplanar Orbits. *Advances in the Astronautical Sciences*, 85:483–502, 1993.

[6] C.K. Carrington and J.L. Junkins. Time-varying potential function control for constrained attitude tracking. *Journal of Guidance, Control, and Dynamics*, 9(1):99–107, 1986.

[7] T. Crain, J. Condon, C. D'Souza, and M. Weeks. Entry Interface Velocity Variation. CEV Flight Dynamics Presentation, May 16 2007.

[8] E. Feron E. Frazzoli, M.A. Dahleh. A randomized attitude slew planning algorithm for autonomous spacecraft. In *AIAA Guidance, Navigation, and Control Conference*, Montreal, Quebec, Canada, Aug. 2001. AIAA 2001-4155.

[9] P. J. Enright and B. A. Conway. Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming. *Journal of Guidance, Control, and Dynamics*, 14(5):981–985, 1991.

[10] R. H. Gooding. A procedure for the solution of lambert's orbital boundary-value problem. *Celestial Mechanics and Dynamical Astronomy*, 48:145–165, 1990.

[11] J. C. Harpold and Jr. C. A. Graves. Shuttle Entry Guidance. *Journal of the Astronautical Sciences*, 27:239–268, July-Sept. 1979.

[12] K. C. Howell, B. T. Barden, R. S. Wilson, and M. W. Lo. Trajectory design using a dynamical systems approach with application to GENESIS. In *Proceedings of the AAS/AIAA Astrodynamics Conference, Sun Valley, Idaho*, pages 1665–1684, 1997.

[13] K. C. Howell and H. J. Pernicka. Numerical Determination of Lissajous Trajectories in the Restriced Three-Body Problem. *Celestial Mechanics*, 41:107–124, 1988.

[14] David G. Hull. *Optimal Control Theory for Applications*. Springer, New York, NY, 2003.

[15] C. T. Hyde, C. E. Foggat, and B. D. Weber. Apollo Experience Report - Abort Planning. Technical Report NASA-TN-D-6847, National Aeronautics and Space Administration, Houston, TX, June 1972.

[16] Roland F. Jaggers. Shuttle powered explicit guidance (peg) algorithm. Technical Report JSC-26122, NASA/JSC Flight Design and Dynamics Division, Nov. 1992.

[17] Ed. James R. Wertz. *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, Norwell, MA, 1978.

[18] D. J. Jezewski. Primer Vector Theory and Applications. NASA Technical Report R-454, NASA Johnson Spaceflight Center, 1975.

[19] Richard P. Kornfeld. On-board autonomous attitude maneuver planning for planetary spacecraft using genetic algorithms. In *AIAA Guidance, Navigation, and Control Conference*, Austin, TX, Aug. 2003.

[20] K. KrishnaKumar, S. Rickard, and S. Bartholomew. Adaptive neurocontrol for spacecraft attitude control. *Neurocomputing*, 9:131–148, 1995.

[21] D. F. Lawden. Impulsive transfer between elliptical orbits. In G. Leitmann, editor, *Optimization Techniques: with applications to aerospace systems*, pages 875–880. Academic Press, Burlington, VT, 1962.

[22] D. F. Lawden. Optimal Transfers Between Coplanar Elliptical Orbits. *Journal of Guidance, Control, and Dynamics*, 15(3):788–791, 1992.

[23] Y.Y. Lin and L.G. Kraige. Enhanced techniques for solving the two-point boundary-value problem associated with the optimal attitude control of spacecraft. *Journal of the Astronautical Sciences*, 37(1):1–15, 1989.

[24] W. H. T. Loh. *Re-Entry and Planetary Entry Physics and Technology*. University of Michigan Press, Ann Arbor, MI, 1980.

[25] B. G. Marchand, K. C. Howell, and R. S. Wilson. Improved Corrections Process for Constrained Trajectory Design in the n-Body Problem. *Journal of Spacecraft and Rockets*, 44(4):884–897, 2007.

[26] Gianmarco Radice Massimo Casasco. Time-varying potential function control for constrained attitude tracking. *Advances in the Astronautical Sciences*, 119(1):555–574, 2004.

[27] Colin R. McInnes. Autonomous path planning for on-orbit servicing vehicles. *Journal of the British Interplanetary Society*, 53:26–38, 2000.

[28] C.R. McInnes. Non-linear control for large angle attitude slew maneuvers. In *Proceedings of the Third International Conference on Spacecraft Guidance, Navigation and Control Systems*, Noordwijk, The Netherlands, Nov. 1996.

[29] Thomas J. Meissen. Space shuttle lambert guidance improvement, scr 92843/93009. Technical Report JSC-49830, NASA/JSC Flight Design and Dynamics Division, Oct. 2006.

[30] A. Miele, M. Ciarcia, and M. W. Weeks. Guidance Trajectories for Spacecraft Rendezvous. *Journal of Optimization Theory and Applications*, 132:377–400, 2007.

[31] J. Neubauer. *Controlling Swarms of Micro-Utility Spacecraft.* PhD thesis, Washington University in St. Louis, August 2002.

[32] C. Ocampo. Finite Burn Maneuver Modeling for a Generalized Spacecraft Trajectory Design and Optimization System. *Annals of the New York Academy of Sciences*, 1017:210–233, 2004.

[33] C. A. Ocampo. An Architecture for a Generalized Spacecraft Trajectory Design and Optimization System. In *Libration Point Orbits and Applications: Proceedings of the Conference, Aiguablava, Spain*, pages 529–571, June 2002.

[34] C. A. Ocampo and R. R. Saudemont. Initial Trajectory Model for a Multi-Maneuver Moon to Earth Abort Sequence. In *AAS/AIAA Space Flight Mechanics Meeting.* Savannah, GA, Feb. 2009. AAS 09-195.

[35] P. A. Penzo. An Analysis of Moon-to-Earth Trajectories. Technical Report NASA-CR-132100, Space Technology Laboratories, Inc., Redondo Beach, CA, Oct. 1961.

[36] A. E. Petropoulos and J. M. Longuski. Shape-Based Algorithm for Automated Design of Low-Thrust, Gravity-Assist Trajectories. *Journal of Spacecraft and Rockets*, 41(5):787–796, 2004.

[37] Walter Murray Philip E. Gill and Margaret H. Wright. *Practical Optimization*. Academic Press, 1981.

[38] Walter Murray Philip E. Gill and Margaret H. Wright. *Practical Optimization*. Academic Press, 1981.

[39] J. Prussing and B. Conway. *Orbital Mechanics*. Oxford University Press, New York, NY, 1993.

[40] David Long R. McHenry, A. Leroy and Roland F. Jaggers. Exoatmospheric generalized guidance for shuttle, 73-fm-168. Technical Report JSC-08664, NASA/JSC Mission Planning and Analysis Division, Dec. 1973.

[41] C. L. Ranieri and C. A. Ocampo. Optimization of Roundtrip, Time-Constrained, Finite Burn Trajectories via an Indirect Method. *Journal of Guidance, Control, and Dynamics*, 28(2):306–314, 2005.

[42] H. M. Robbins. An analytical study of the impulsive approximation. *AIAA Journal*, 4(8):1417–1423, 1966.

[43] J. Savage, E. Marquez, J. Pettersson, N. Trygg, A. Petersson, and M. Wahde. Optimization of waypoint-guided potential field navigation using evolu-

tionary algorithms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3463 – 3468, Sendai, Japan, Sept. - Oct. 2004.

[44] S. K. Scarritt, B. G. Marchand, A. J. Brown, W. H. Tracy, and M. W. Weeks. A finite burn linear targeting algorithm for autonomous path planning and guidance. *Journal of Guidance, Control, and Dynamics*, In Revision.

[45] M D Shuster. A survey of attitude representations. *Journal of the Astronautical Sciences*, 41(4):439–517, Oct.-Dec. 1993.

[46] C. W. Smith. An Onboard Targeting Algorithm with Earth-Return Applications. Master's thesis, University of Texas at Austin, August 2008.

[47] S. A. Stanton. Optimal Orbital Transfer Using a Legendre Pseudospectral Method. Master's thesis, Massachusetts Institute of Technology, August 2003.

[48] N. J. Strange and J. M. Longuski. Graphical Method for Gravity-Assist Trajectory Design. *Journal of Spacecraft and Rockets*, 39(1):9–16, 2002.

[49] F. T. Sun. Analytic solution for optimal two-impulse 180° transfer between noncoplanar orbits and the optimal orientation of the transfer plane. *AIAA Journal*, 7(10):1898–1904, 1969.

[50] M. Swartwout, S. Scarritt, and J. Neubauer. Potential Function controllers for Proximity navigation of Underactuated Spacecraft. *Advances in the Astronautical Sciences*, 128:267–286, 2007.

[51] Andrew Tatsch and Norman Fitz-Coy. Dynamic artificial potential function guidance for autonomous on-orbit servicing. In *6th International ESA Conference on Guidance, Navigation and Control Systems*, Loutraki, Greece, Oct. 2005.

[52] P. Teofilatto and E. De Pasquale. A fast guidance algorithm for an autonomous navigation system. *Planetary and Space Science*, 46(11/12):1627–1632, 1998.

[53] Dennis W. Brown Timothy J. Brand and John P. Higgins. Unified powered flight guidance, space shuttle gn&c equation document no. 24, revision 1. Technical report, Charles Stark Draper Laboratory, Apr. 1974.

[54] David A. Vallado. *Fundamentals of Astrodynamics and Applications, 2nd Edition*. Microcosm Press, El Segundo, CA, 2004.

[55] N. X. Vinh, A. Busemann, and R. D. Culp. *Hypersonic and Planetary Flight Mechanics*. University of Michigan Press, Ann Arbor, MI, 1980.

[56] N. X. Vinh, S. H. Kuo, and C. Marchal. Optimal time-free nodal transfers between elliptical orbits. *Acta Astronautica*, 17(8):875–880, 1988.

[57] M. W. Weeks and T. Crain. Description of a TransEarth Midcourse Targeting Algorithm for CEV. Doc. No. FltDyn-CEV-07-012, Feb. 8th 2007.

[58] M. W. Weeks, B. G. Marchand, C. W. Smith, and S. Scarritt. Onboard Autonomous Targeting for the Trans-Earth Phase of Orion. *Journal of Guidance, Control, and Dynamics*, May-June.

[59] M. W. Weeks, B. G. Marchand, C. W. Smith, and S. K. Scarritt. Design of the Onboard Autonomous Targeting Algorithm for the Trans-Earth Phase of Orion. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Honolulu, HI, Aug. 2008. AIAA 2008-7262.

[60] R. S. Wilson, B. T. Barden, K. C. Howell, and B. G. Marchand. Summer Launch Options for the Genesis Mission. *Advances in the Astronautical Sciences*, 109:77–94, 2002.

[61] R. S. Wilson and K. C. Howell. Trajectory Design in the Sun-Earth-Moon System Using Lunar Gravity Assists. *Journal of Spacecraft and Rockets*, 35(2):191–198, 1998.

[62] J. D. Yencharis, R. F. Wiley, R. S. Davis, Q. A. Holmes, and K. T. Zeiler. Apollo Experience Report - Development of Guidance and Targeting Techniques for the Command Module and Launch Vehicle. Technical Report NASA-TN-D-6848, National Aeronautics and Space Administration, Houston, TX, June 1972.

[63] Paul Zarchan. *Tactical and Strategic Missile Guidance, 5th Edition.* AIAA, Reston, VA, 2007.

[64] A.T. Zaremba. An adaptive scheme with parameter identification for spacecraft attitude control. In *IEEE American Control Conference*, pages 552–556, Jun 1997.

[65] S. Zimmer and C. Ocampo. Use of Analytical Gradients to Calculate Optimal Gravity-Assist Trajectories. *Journal of Guidance, Control, and Dynamics*, 28(2):324–332, 2005.

Index

Vita

Sara Kathryn Scarritt was born in 1983 in Alexandria, Virginia, and grew up in Birmingham, Alabama. Sara attended Princeton University, where she majored in Mechanical and Aerospace Engineering, graduating in 2005 with a Bachelor of Science in Engineering (B.S.E.) degree and a minor in Robotics and Intelligent Systems. In 2007, Sara earned her Master of Science in Aerospace Engineering from Washington University in St. Louis. During the summer and fall of 2007, she worked as an intern at the Air Force Research Laboratory at Kirtland AFB, NM. Sara enrolled at the University of Texas at Austin in January 2008 to pursue her Ph.D. She will begin work at NASA's Johnson Space Center in the summer of 2012.

Permanent address: scarritt@alumni.princeton.edu

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.